

Open Source Security Tools for Information Technology Professionals

School of Professional Studies (SPS)

The City University of New York (CUNY)

Aron Trauring
Adjunct Professor
CEO, Zoteca

Class 10 November 28th, 2005

Class Agenda

6:30-6:45 Article of the Week

6:45-7:00 Knoppix/Ubuntu Live

7:00-7:30 Network Sniffer Lab Part II

7:30-8:00 Network Intrusion Detection Systems — Part I

8:00-8:15 Break

8:15-8:45 Network Intrusion Detection Systems — Part II

8:45-9:30 NIDS Lab — Part I

Instructor

Aron Trauring – Adjunct Professor, CUNY SPS / CEO, Zoteca

Email: atrauring@zoteca.com

Personal Website: <http://aronst.org/>

Zoteca Corporate Website: <http://www.zoteca.com/>

Sniffer Lab

Ethereal

- Like tcpdump with a friendly graphical interface
- Offers many more analytical and statistical options.
- Output is much easier to read and understand than the raw packet captures of Tcpdump.
- Can interpret over 300 different network protocols, which covers just about every network type ever invented.
- More physical network formats are supported.
- Output can be saved as plain text or in Postscript format.
- A rich display filter mode. This includes the ability to highlight certain packets in color.
- There is a filter creation GUI to walk you through the process of creating filters easily.
- The ability to follow a TCP stream and view the content in ASCII.
- The ability to work with dedicated hardware
- The ability to save sessions in multiple formats.
- A command-line terminal mode. (so can run on servers without GUI - can dump file for analysis in GUI)
- Can also use serve as a general network analysis tool.

Using Ethereal

- Choose Capture --> Start
- Choose Prepare
- Can do in real time or capture for later analysis

Capture Options

- Interface — Picks the interface to capture from the pull-down menu.
- Ethereal automatically senses all the available interfaces and lists them.
- You can also choose to capture from all interfaces at once, just like Tcpdump
- Limit each packet to x bytes — Sets a maximum size for the packets captured.
- Use this if you fear some of the packets may be very large and you don't want to overload your machine.
- Capture packets in promiscuous mode — on by default.
- Filter — creates a filter using tcpdump-style expressions.
- Can name the filter so can the use in future sessions
- Capture file(s) — if want to read from a file rather than capture live data.

- Display options — these are disabled by default, but enable them if you want to watch the packets scroll by in real time.
- Not recommended for busy network or slow machine is slow,
- Capture limits — automatic stop
- After x number of packets or kilobytes of data have been captured,
- After x number of seconds have elapsed.
- Name resolution — specify whether you want Ethereal to resolve names at various levels of the network model.
- Enabling all of these, especially DNS, can slow down your capture significantly.

Packet List Window

- The top third of the screen is where the packet stream is displayed in order of receipt,
- Can sort this in just about any way by clicking on the headings.
- Packet number — Assigned by Ethereal
- Time — The time the packet was received, set from the elapsed time from the start of the capture session.
- Alternately, this can be configured to show the clock time, the clock time and date, or even the time between packets (this is helpful for network performance analysis).
- Source address — Where the packet came from. This is an IP address on IP networks.
- Destination address — Where the packet is going to, also usually an IP address.
- Protocol — The level 4 protocol that the packet is using
- Info — Some summary information about the packet, usually a type field.

Packet Detail Window

- Goes into more detail on each packet that is highlighted.
- Arranged in an order that basically conforms to the OSI model, so the first item listed is detail on the data link layer, and so on.
- The little pluses can be expanded to show even more information on each level. It is amazing how much detail you can see on each packet.

Packet Content Window

- Actual packet contents, in both hexadecimal and translated into ASCII where possible.
- Binary files will still look like garbage, as will encrypted traffic, but anything in clear text will appear.
- Highlights the power (and danger) of having a sniffer on your network.

Preferences

- Allows you to change layout
- Allows you to modify and add headings
- Allows you to adjust ports and such for protocols

Display Filters

- Choose Expression
- Choose protocol and filter type
- Choose apply

Ethereal Session Statistics Window

- Displays protocol statistic
- You can stop your session at any time by clicking Stop
- If you set a limit in the options, it will automatically stop when it reaches it.

Analyze

- Follow TCP stream— shows connection traffic “conversation”

Statistics

tethereal

- `tethereal -w <filename>`
- `ethereal -r <filename>`

ntop

- `ntop -u root`
- first time asks for password
- run in browser `http://localhost:3000/`

Network Intrusion Detection Systems (IDS)

What is an IDS?

- Modified network sniffer
- Sees all traffic on the network
- Tries to sense potential bad traffic
- Sends alert when bad traffic is found

How Does and IDS Work?

1. Compares network traffic to a database of known bad activity (Network IDS)
2. Check integrity of key system files (Host IDS)

Purpose of an IDS

- To detect traffic that gets through a misconfigured firewall
- To detect malicious traffic on ports open to legitimate applications
- To detect attacks coming from inside (behind firewall)

IDS Signature Example

- Nimda and Code Red use the `cmd.exe` attack against IIS
- With IIS having admin privileges, execute `cmd.exe` in a writable directory
- Exploits an IIS buffer overflow in IIS module ISAPI
- No reason for legitimate user to be copying this file over network
- If see network traffic with name "`cmd.exe`" can flag as intrusion

The False Positive Problem

- Generating an alert for normal traffic
- Database contains thousands of signatures
- Can't know what "normal" is on your network
- Default settings may be overly sensitive
- Too many false positives makes IDS useless

Network Monitoring System (NMS) False Positives

- An NMS keeps track of network traffic
- Generate polling and discovery activity
- Use SNMP protocol but may also use ICMP pings
- An NMS can generate thousands of alerts per hour on the IDS
- Set up IDS to ignore traffic to and from NMS

Network Scanning False Positives

- Nmap or Nessus can also set off the IDS
- Best solution is to shut down IDS when doing scanning
- Otherwise alert database will be skewed with false data

User Activity False Positives

- IM and P2P are flagged
- Decide whether or not to ban
- Either comment out flagging rules (leaving yourself exposed) or enforce bans

Application Software False Positives

- Microsoft Exchange behaves just like Nimda worm
- It's webmail interface copies over system files with .eml extension
- Either comment out flagging rules (leaving yourself exposed) or ban Exchange

Long Web Authentication Strings False Positives

- Using long login strings can be a buffer overflow attempt
- Can also be a web application cramming in alot of information
- Fix your applications if you can

Database Authentication Activity False Positives

- IDS look for alot of authentication activity on RDBMS
- Assumption is production databases shouldn't often authenticate
- Development databases can cause problems
- Disable alerts in development environments
- Proper IDS Configuration

Customize Settings for your LAN environment

- Shut off categories not relevant (e.g. Unix signatures in an all Microsoft environment)
- Shut off policies not relevant (e.g. IM or P2P as noted above)
- Exempt problematic hosts (e.g. NMS, development databases, admin machine)
- Reducing false positives does increase risks but makes IDS more useful

Constantly Tune

- Analyze alerts and look for more settings that generate false positives
- Get a feel for “normal” traffic on your network
- May take several months to fully tune

Analyze Data

- Logging data can be too easily ignored
- Email alerts can be overwhelming
- Best alternative is to use a database logging and analysis tool (e.g. ACID)

Main Problems with IDS

- False positive noise
- Enumerate badness fallacy
- Open barn door approach
- Don't overly rely on IDS

Anomalous Activity-Based IDS

- Attacks “enumerate badness” approach
- Monitors normal system activity
- Alerts for unusual activity that is outside the norm
- Takes a long time to learn what is “normal”
- Can be exploited by internal users who know network well

Intrusion Prevention Systems (IPS)

- Attacks “open barn door” approach
- Takes action as alerts are generated
- Write on-the-fly custom firewall rules
- Block IP addresses
- Interrogate or counter-attack offending system
- Many contend just buzzword — most likely will migrate to firewall

Snort Hardware

- Snort should be run on a stand-alone machine
- Minimal install
- No X-windows or other services
- Several gigabytes of hard disk space in separate partition (in case log overflows)
- 500MHZ Intel machine is fine

NIDS Placement on LAN

- Place behind firewall so can see what firewall let's through
- Windows networking generates many alerts so needs to be tuned
- In switched environment can place in line between firewall and switch
- Can also place a hub between firewall and internal switch to avoid single point of failure
- Better option in larger networks is to mirror all ports to a monitor port (more expensive solution)
- More information <http://www.snort.org/docs/faq/1Q05/node9.html>

NIDS Placement on DMZ

- Servers that are most exposed
- Many alerts due to port scanning etc.
- Focus on application-specific alerts and not reconnaissance
- Switching issues same as on LAN

NIDS Placement Between Firewall and ISP

- Can see everything attacking both public servers and internal LAN
- Won't see internal traffic
- Same reconnaissance traffic issue as in DMZ
- Same switching issue as in DMZ

NIDS Lab

Snort

- GPL licensed CLI tool
- Commercial company SourceFire provides tested rules, support and products
- Purchased by Check Point

Other FOSS IDS

- Bro
- Prelude

Invoking Snort

- `snort <options> <expression>`
- Expression similar to `tcpdump` selectors

Snort Sniffer Mode

- similar to `tcpdump` or `tethereal`
- `-v` — displays TCP/IP headers
- `-d` — displays application layer data
- `-e` — displays link layer headers

Snort Logging Mode

- logs all packets sniffed
- `-l <logpath directory>` — logs into subdirectories by IP address
- `-h <home network (local IP range)>` — logs into subdirectories by non-local IP address
- `-b` — logs into binary file that can be analyzed by `ethereal` or `tcpdump`

Snort IDS Mode

- logs only packets that are suspicious or warrant further attention
- `-c <config file>` — configuration file sets parameters for logging
- In Ubuntu and others: `/etc/snort/snort.conf` is default config file
- `/var/log/snort` is default logging directory
- can leave off `-vde` switches which slows down and may cause packets to drop

Snort Alert Modes

- `-A full` — full alert information. Default when nothing is specified
- `-A fast` — logs only packet header and alert type (useful on fast networks)
- `-M <workstation>` — use smb to send to Windows pop-up service
- `-s` — send to Unix syslog
- send to database for later analysis

Example Snort Session

- `snort -A full -c /etc/snort/snort.conf -b`
- in other window run `nmapfe`
- Control-C after a while
- `ls -al /var/log/snort`
- `ethtool -r /var/log/snort/<logfile>`

Snort Configuration — Home Network

- `var HOME_NET <addresses>`
- Addresses comma-separated list of local network e.g. `192.168.1.0/24`
- `var HOME_NET $<interface>`
- Takes IP and mask from interface configuration e.g. `eth0`
- `EXTERNAL_NET` can also be defined
- default for both is `any`

Snort Configuration — Internal Servers

- Define servers you have running on your network
- Can specify ports so only registers attacks on open ports
- Limits false positives

Snort Configuration — Decoders and Pre-processors

- Run on traffic before it passes through rule sets
- For proper formatting or types of traffic easier to deal with as a class
- Example: decoder to reassemble fragmented packets so properly formatted
- Example: pre-processor for port scanning traffic which is high-volume and better dealt with en masse
- Only change configuration as gain more experience with the tool

Snort Configuration — Output Modules

- Used for managing output
- Three modules: syslog, database, unified (binary format)
- `output <module name>: <configuration options>`
- See documentation for details

Snort Configuration — Rule Sets

- Rule sets are in `rules` directory
- In config file can turn on/off whole rule sets via deleting/adding comment #

Snort Configuration — Individual Rules

- Go to rules directory and edit rules files
- Can turn on/off individual rule via deleting/adding comment #
- Better to work on individual rules than whole set if relevant in anyway

Rule Classes	Descriptions
attack-responses rules	These are alerts for common response packets after an attack is successful. They should rarely report false positives and should be left on in most cases.
backdoor rules	These are common signs a backdoor or Trojan horse program is in use. They will rarely be false positive.
bad-traffic rules	These rules represent nonstandard network traffic that should not typically be seen on most networks.
chat rules	Look for standard sign-ons for many popular chat programs. If chat is allowed explicitly or implicitly, then these alerts should be turned off. Also, note that these are not silver bullets for chats and will not detect all types of chat traffic. Still, they can be helpful in ferreting out the worst offenders.
ddos rules	Look for standard distributed denial of service types of attacks. On a DMZ and WAN, these alerts don't serve much purpose, because if you are under a distributed denial of service you will probably know it right away. However, they can be very useful inside the LAN to see if you have zombie machine participating unknowingly in a DDOS attack on another network.

Rule Classes	Descriptions
dns rules	Look for some standard exploits against DNS servers. If you aren't running your own DNS, you can turn these off.
dos rules	Similar to the ddos.rule set above.
experimental rules	These are turned off by default. These are generally used only for testing new rules until they are moved into one of the other categories.
exploit rules	These are for standard exploit traffic and should always be enabled.
finger rules	These rules flag traffic having to do with finger servers. If you are not running finger anywhere, you could probably turn these off. However, finger servers often are running hidden from the system administrator, so you could leave these on as they shouldn't generate false positives if you don't have any.
ftp rules	Same as finger rules but looking for FTP exploits. Again, there is no harm in leaving them enabled even if you don't have FTP servers since it will alert you to any rogue FTP servers you may have.
icmp-info rules	These rules track the use of ICMP messages crossing your network, for example, pings. These are often the cause of false positives, and you may want to disable the whole lot unless you want to keep a close eye on ICMP traffic on your network. Another class for known bad ICMP traffic, icmp rules catches ports scans and the like.
icmp rules	Cover bad or suspicious ICMP traffic such as port scans, and are less likely to generate false positives. However, it is possible they will be triggered often on a busy network with lots of diagnostic services running.
imap rules	Rules regarding the use of Internet Message Access Protocol (IMAP) on your network.
info rules	Trap miscellaneous error messages on your network from Web, FTP, and other servers.
local rules	You add your own custom signatures for your network in this file. This file is empty by default. See the documentation for information on writing a custom Snort rule.
misc rules	Rules that don't fit under one of the other categories or don't warrant their own sections are in this file. An example would be older alerts like Gopher server exploits.

Rule Classes	Descriptions
multimedia rules	Track usage of streaming video type software. If you allow streaming video applications or use video conferencing on your network, then you will want to disable these rules.
mysql rules	Watch for administrator access and other important files in a MySQL database. If you don't run MySQL, then you can probably disable these alerts. Also, if your MySQL database is under development, these might trigger a lot of false positives.
Netbios rules	This class of rules alerts you to various NetBIOS activity on your LAN. Some of them are obvious exploits. However, others, such as the NULL session alerts, may happen normally on a Windows LAN. You will have to play with this section to figure out the rules that are appropriate for your LAN.
nntp rules	News server-related rules. If you don't run network news on your servers, you can probably turn these off.
oracle rules	Oracle database server rules. Again, if you don't run it, turn it off.
other-ids rules	These rules are related to exploits on other IDS manufacturers' boxes. Chances are that you don't have any NIDS on your LAN, but if you do, leave these on.
p2p rules	Rules governing peer-to-peer file sharing software use. These rules will create alerts during normal use of these products, so if you allow this software then you will need to turn these off.
policy rules	This file contains various alerts relating to allowed activity on the LAN, such as Go-to-my-pc and other programs. You should review these and enable only the ones that apply to your internal policies.
pop2 / pop3 rules	Both files to mail servers. Most companies, if using POP, will be using a POP3 server. If you have either of these types of servers, leave these rules on; if not, disable them.

Rule Classes	Descriptions
porn rules	These are some rudimentary traps for pornography-related Web surfing. These are by no means a replacement for a good content-filtering system, but can catch some of the more egregious violators.
rpc rules	This class handles remote procedure call (RPC) alerts. Even though you may not think you are running any of these services, they often run as part of other programs, so it is important to be aware when this is happening on your LAN. RPC can enable remote code execution and is often used in Trojans and exploits.
rservices rules	Track use of various remote services programs, such as rlogin and rsh. These are insecure services in general, but if you have to use them, they can be tracked closely with this rule set.
scan rules	Alert you to use of port scanning programs. Ports scans are a good indication of illicit activity. If you use port scanners, you will want to either turn off Snort during those times or disable the particular rule for your scanner machine.
shellcode rules	This class looks for packets containing assembly code, low-level commands also known as shell code. These commands are often integral to many exploits such as buffer overflows. Catching a chunk of shell code flying by is often a pretty good indication that an attack is underway.
smtp rules	Govern alerts for mail server use on the LAN. This section will need some fine-tuning, as many normal mail server activities will set off rules in this section.
sql rules	Rules for various SQL database programs. If you don't run any databases you can turn these off, but it's not a bad idea to leave them on just in case there are SQL databases running that you don't know about.
telnet rules	Track Telnet use on the network. Telnet is often used on routers or other command line devices, so it is a good thing to track even if you don't run Telnet on your servers.

Rule Classes	Descriptions
tftp rules	TFTP (trivial FTP) is an alternate FTP server often run on routers. It can be used to upload new configurations and therefore is worth keeping an eye on.
virus rules	Contain signatures of some common worms and viruses. This list is not complete and is not maintained regularly. It is not a replacement for virus scanning software but can catch some network-aware worms.
web-attacks rules web-cgi rules web-client rules web-coldfusion rules web-frontpage rules web-iis rules web-php rules	All these classes refer to various kinds of suspicious Web activity. Some are generic, such as the web-attacks class. Others, like web-iis and web-frontpage, are specific to a particular Web server platform. However, even if you don't think you run a Microsoft Web server or use PHP, it is worth leaving them all running to uncover any of this kind of activity on your LAN you may be unaware of. You will have to do some fine-tuning of the rule sets, especially if your Web servers are in active development.
X11 rules	Track the use of the X11 graphical environment on your network.

Webmin Snort