Fachhochschule Nordwestschweiz
Hochschule für Technik

Tools for penetration tests

Carlo U. Nicola, HT FHNW

With extracts from documents of :

Google; Wireshark; nmap; Nessus.

What is a penetration test?

Goals:

1. Analysis of an IT-environment and search for exploitable vulnerabilities
2. “Simulation" of the behaviour of an attacker
3. The effort spent is an indication of the effort an attacker would have to spend to achieve the same results (i.e. to uncover vulnerabilities)

Phases of a penetration test:

(1) Foot-printing: collecting relevant information about the target environment

(2) Scanning: examine the target networks and hosts in more detail

(3) Analysis : identify the systems that will be analyzed in detail during the following phase

(4) Exploitation: demonstrate proof-of-concept or "real“ exploits

(5) Reporting: Prepare a written report and an oral presentation, including concrete recommendations

Foot-printing

Foot-printing means collecting the profile of the target openly accessible in Internet.

What information are we interested in?

A not exhaustive list:

1. Domain names,
2. Contact persons,
3. Interesting pairs (hostnames, IP addresses) connected with the company’s name,
4. IP address blocks,
5. Internal system configurations

Foot-printing: Domain names (1) + (2)

The following tools are most useful in this phase:

1. Google (simple type the company domain name and analyse all additional domains linked to it.)
2. `whois` as command on *nixes or as service via links: `whois.iana.org` (query the TLD (Top Level Domain) field i.e. `ch` for Switzerland. For `.li` and `.ch` TLD http://www.nic.ch gives the most authoritative answers.
3. Browsing the company page. Use the search function within the company page with the contact info you extracted from the `whois` search.

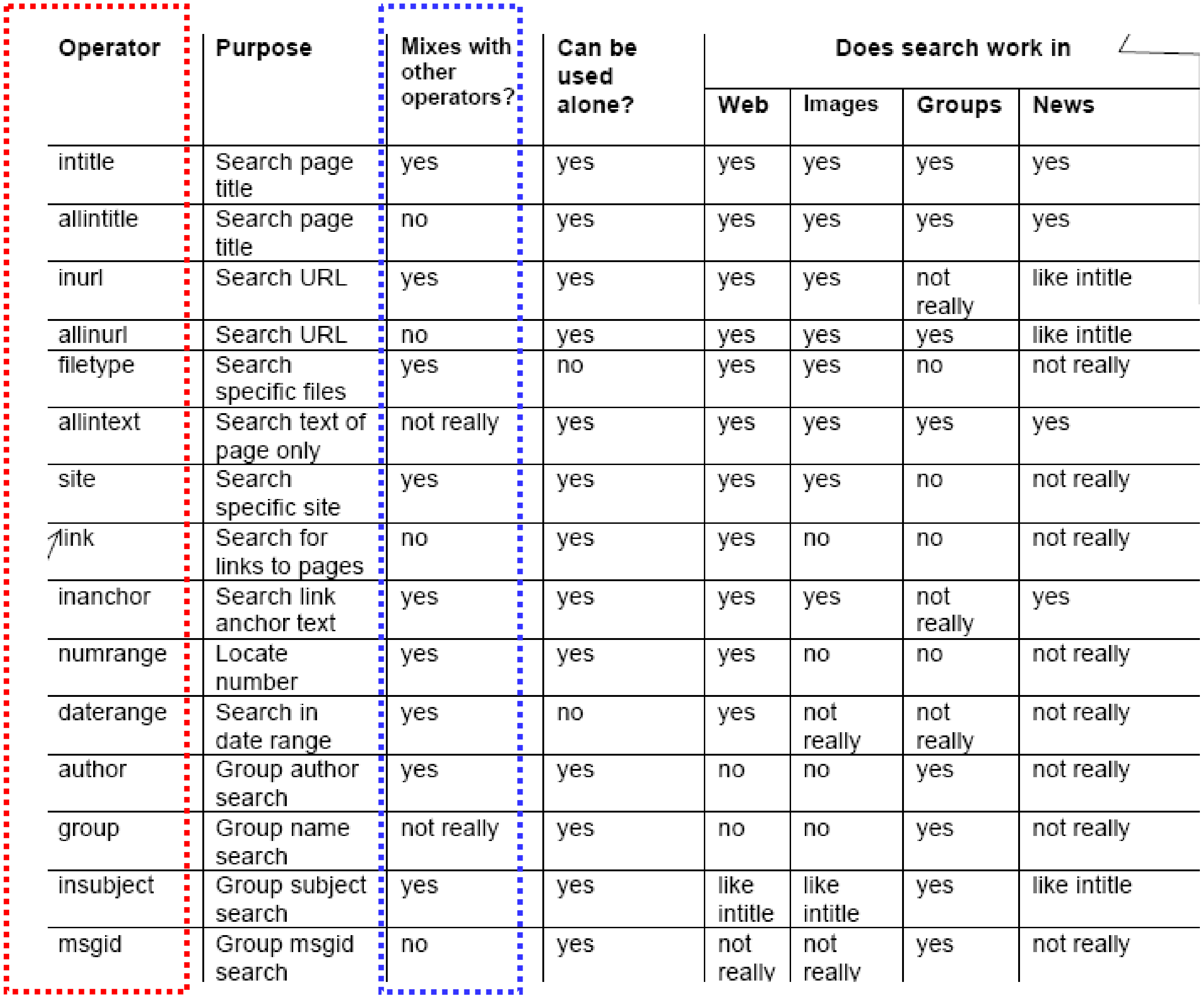
An versatile test tool: Google

Google in the early foot-print phase

Below are some questions that are best answered using the operators Google internally uses for its searches:

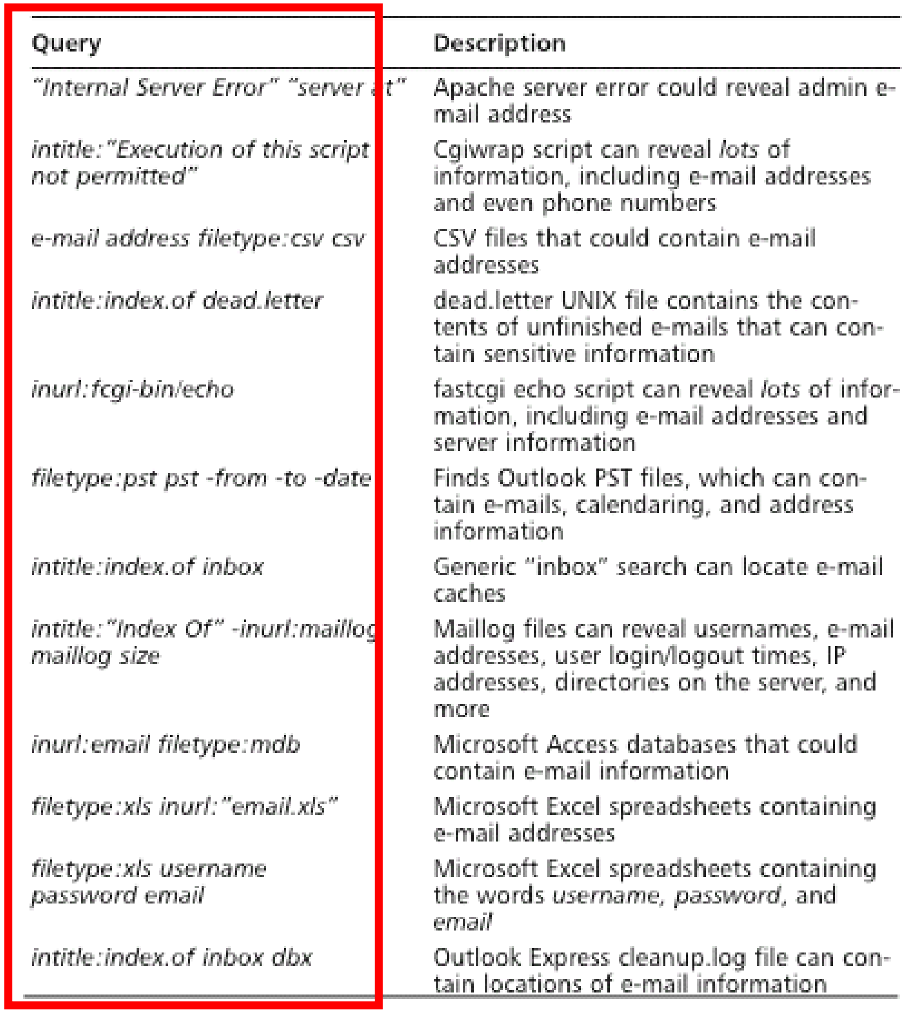
1. We want to know email addresses and important documents used in the company’s network;
2. We want to check whether important information no longer available is still around perhaps cached somewhere else by Google;
3. We want to map the less obvious connections within the system's domain we are testing;
4. We want to probe which other domains link to the system under investigation;
5. …

The Google main operators

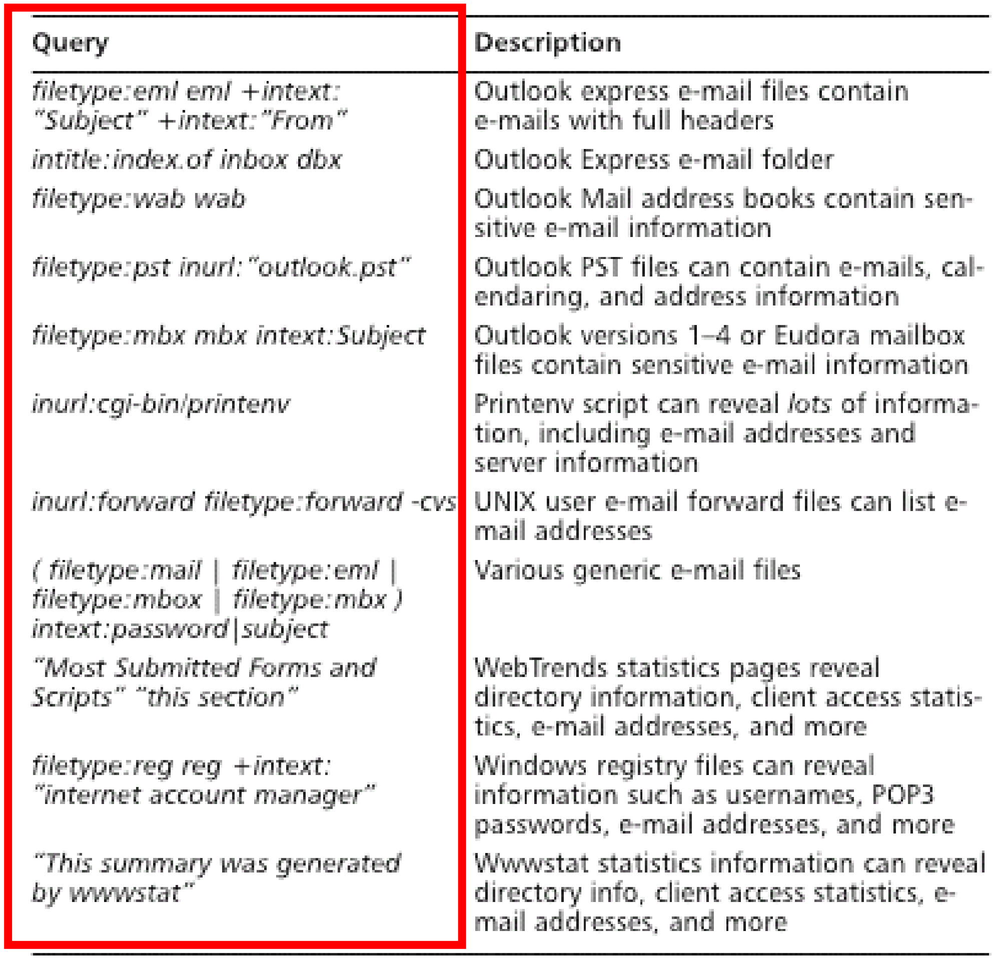


Operator	Purpose	Mixes with other operators?	Can be used alone?	Does search work in			
				Web	Images	Groups	News
intitle	Search page title	yes	yes	yes	yes	yes	yes
allintitle	Search page title	no	yes	yes	yes	yes	yes
inurl	Search URL	yes	yes	yes	yes	not really	like intitle
allinurl	Search URL	no	yes	yes	yes	yes	like intitle
filetype	Search specific files	yes	no	yes	yes	no	not really
allintext	Search text of page only	not really	yes	yes	yes	yes	yes
site	Search specific site	yes	yes	yes	yes	no	not really
link	Search for links to pages	no	yes	yes	no	no	not really
inanchor	Search link anchor text	yes	yes	yes	yes	not really	yes
numrange	Locate number	yes	yes	yes	no	no	not really
daterange	Search in date range	yes	no	yes	not really	not really	not really
author	Group author search	yes	yes	no	no	yes	not really
group	Group name search	not really	yes	no	no	yes	not really
insubject	Group subject search	yes	yes	like intitle	like intitle	yes	like intitle
msgid	Group msgid search	no	yes	not really	not really	yes	not really

Email addresses



Query	Description
"Internal Server Error" "server at"	Apache server error could reveal admin e-mail address
intitle:"Execution of this script not permitted"	Cgiwrap script can reveal *lots* of information, including e-mail addresses and even phone numbers
e-mail address filetype:csv csv	CSV files that could contain e-mail addresses
intitle:index.of dead.letter	dead.letter UNIX file contains the contents of unfinished e-mails that can contain sensitive information
inurl:fcgi-bin/echo	fastcgi echo script can reveal *lots* of information, including e-mail addresses and server information
filetype:pst pst -from -to -date	Finds Outlook PST files, which can contain e-mails, calendaring, and address information
intitle:index.of inbox	Generic "inbox" search can locate e-mail caches
intitle:"Index Of" -inurl:maillog maillog size	Maillog files can reveal usernames, e-mail addresses, user login/logout times, IP addresses, directories on the server, and more
inurl:email filetype:mdb	Microsoft Access databases that could contain e-mail information
filetype:xls inurl:"email.xls"	Microsoft Excel spreadsheets containing e-mail addresses
filetype:xls username password email	Microsoft Excel spreadsheets containing the words *username*, *password*, and *email*
intitle:index.of inbox dbx	Outlook Express cleanup.log file can contain locations of e-mail information



Query	Description
filetype:eml eml +intext: "Subject" +intext:"From"	Outlook express e-mail files contain e-mails with full headers
intitle:index.of inbox dbx	Outlook Express e-mail folder
filetype:wab wab	Outlook Mail address books contain sensitive e-mail information
filetype:pst inurl:"outlook.pst"	Outlook PST files can contain e-mails, calendaring, and address information
filetype:mbx mbx intext:Subject	Outlook versions 1–4 or Eudora mailbox files contain sensitive e-mail information
inurl:cgi-bin/printenv	Printenv script can reveal *lots* of information, including e-mail addresses and server information
inurl:forward filetype:forward -cvs	UNIX user e-mail forward files can list e-mail addresses
(filetype:mail | filetype:eml | filetype:mbox | filetype:mbx) intext:password|subject	Various generic e-mail files
"Most Submitted Forms and Scripts" "this section"	WebTrends statistics pages reveal directory information, client access statistics, e-mail addresses, and more
filetype:reg reg +intext: "internet account manager"	Windows registry files can reveal information such as usernames, POP3 passwords, e-mail addresses, and more
"This summary was generated by wwwstat"	Wwwstat statistics information can reveal directory info, client access statistics, e-mail addresses, and more

Information in cache

If the desired information is not available you can always try to get a cached version. Take the URL and add `&strip=1` as the example below shows:

Original:

```
http://64.233.187.104/search?q=cache:Z7FntxDMrMIJ:www.phrack.org/
hardcover62/+phrack+hardcover62&hl=en
```

Cached text only:

```
http://64.233.187.104/search?q=cache:Z7FntxDMrMIJ:www.phrack.org/

hardcover62/+phrack+hardcover62&hl=en&lr=&strip=1
```

Rough network mapping with Google

How is the domain `www.fhnw.ch` structured?

We want of course to filter out the most obvious combinations, for example all those containing `www.`

```
site:fhnw.ch -site:www.fhnw.ch
```

Or more explicit:

```
site:fhnw.ch -inurl:www.fhnw.ch
```

External links

How is the domain `www.fhnw.ch` linked with other domains?

```
link:www.fhnw.ch
```

But if you type:

```
link:www.fhnw.ch -site:fhnw.ch
```

`link` is treated like a normal search test.

Vulnerabilities: Google Dorks (1)

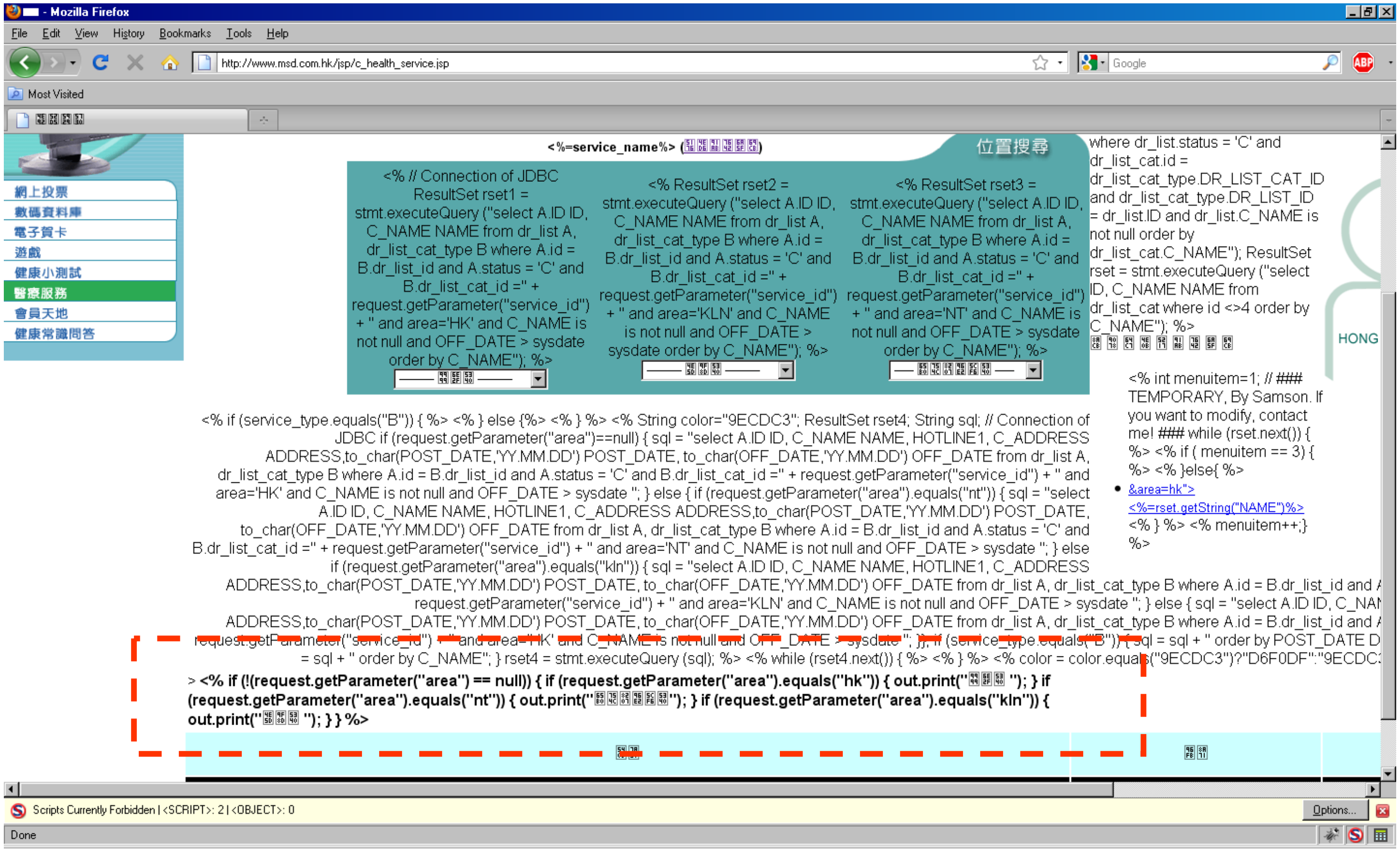
```
<Dork> filetype:php inurl:index.php inurl:"module=subjects"
       inurl:"func=*" (listpages| viewpage | listcat)
</Dork>
<Category>Advisories and Vulnerabilities</Category>
<Query> filetype:php inurl:index.php inurl:"module=subjects"
        inurl:"func=*" (listpages| viewpage | listcat)
</Query>
<Comment> Reportedly the PostNuke Modules Factory Subjects
          module is affected by a remote SQL injection
          vulnerability.
          http://securityfocus.com/bid/11148/discussion/
</Comment>
```

Vulnerabilities: Google Dorks (2)

```
<Dork> "Online Store - Powered by ProductCart"</Dork>
<Category>Advisories and Vulnerabilities</Category>
<Query>"Online Store - Powered by ProductCart"</Query>
<Comment> ProductCart is "an ASP shopping cart that combines
          sophisticated ecommerce features with time-saving
          store management tools and remarkable ease of use.
          It is widely used by many e-commerce sites".
          Multiple SQL injection vulnerabilities have been
          found in the product, they allow anything from
          gaining administrative privileges (bypassing the
          authentication mechanism), to executing arbitrary
          code.
          http://www.securityfocus.com/bid/8105
         (search SF for more)
 </Comment>
```

Google : SQL vulnerability

Query: "executeQuery(“ ".getParameter("

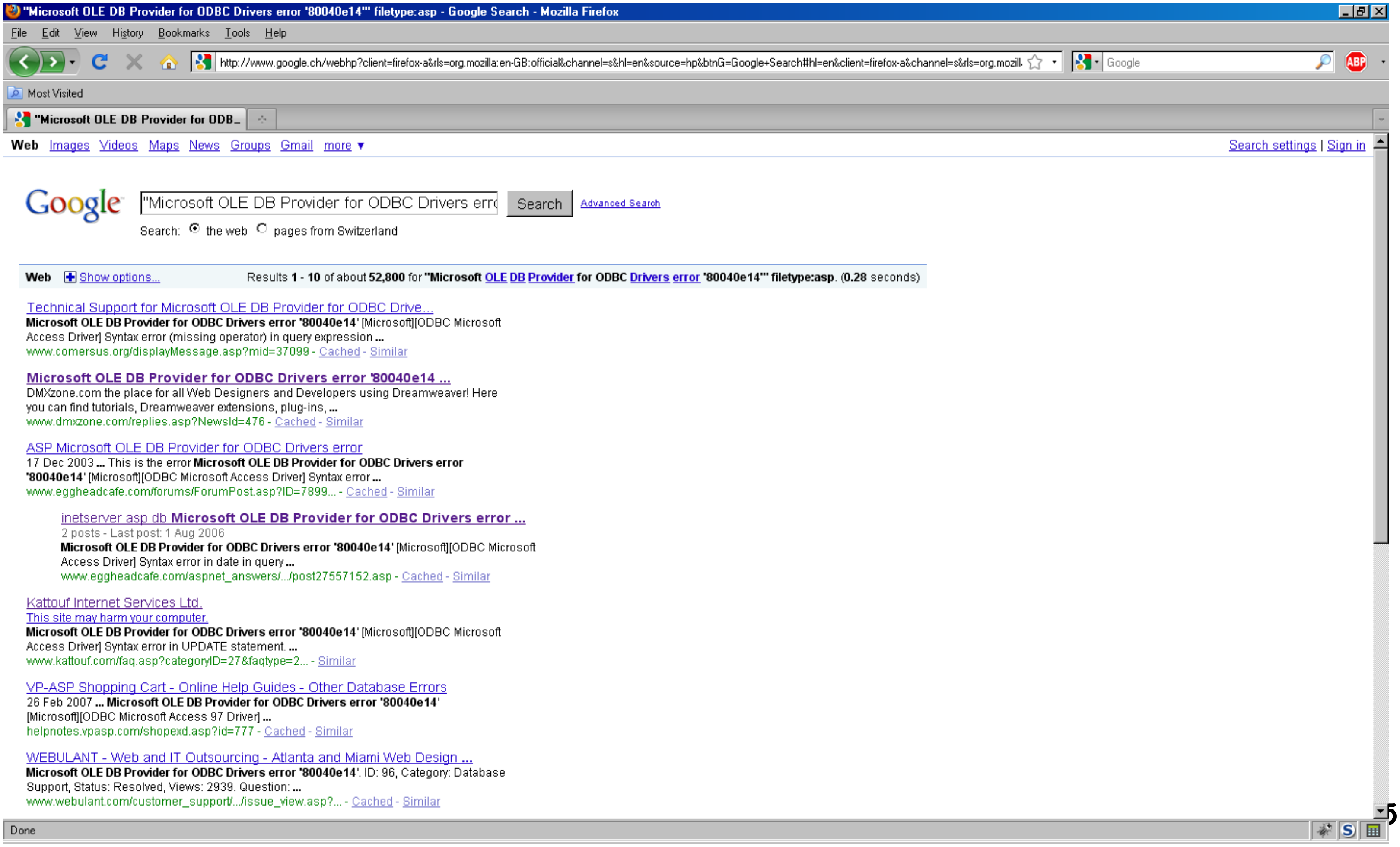

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.msd.com.hk/jsp/c_health_service.jsp
Google
Most Visited
網上投票
數碼資料庫
電子賀卡
遊戲
健康小測試
醫療服務
會員天地
健康常識問答
位置搜尋
HONG
Scripts Currently Forbidden | <SCRIPT>: 2 | <OBJECT>: 0
Options...
Done

```
<%=service_name%> ()

<% // Connection of JDBC ResultSet rset1 = stmt.executeQuery ("select A.ID ID, C_NAME NAME from dr_list A, dr_list_cat_type B where A.id = B.dr_list_id and A.status = 'C' and B.dr_list_cat_id =" + request.getParameter("service_id") + " and area='HK' and C_NAME is not null and OFF_DATE > sysdate order by C_NAME"); %>

<% ResultSet rset2 = stmt.executeQuery ("select A.ID ID, C_NAME NAME from dr_list A, dr_list_cat_type B where A.id = B.dr_list_id and A.status = 'C' and B.dr_list_cat_id =" + request.getParameter("service_id") + " and area='KLN' and C_NAME is not null and OFF_DATE > sysdate order by C_NAME"); %>

<% ResultSet rset3 = stmt.executeQuery ("select A.ID ID, C_NAME NAME from dr_list A, dr_list_cat_type B where A.id = B.dr_list_id and A.status = 'C' and B.dr_list_cat_id =" + request.getParameter("service_id") + " and area='NT' and C_NAME is not null and OFF_DATE > sysdate order by C_NAME"); %>

where dr_list.status = 'C' and dr_list_cat.id = dr_list_cat_type.DR_LIST_CAT_ID and dr_list_cat_type.DR_LIST_ID = dr_list.ID and dr_list.C_NAME is not null order by dr_list_cat.C_NAME"); ResultSet rset = stmt.executeQuery ("select ID, C_NAME NAME from dr_list_cat where id <>4 order by C_NAME"); %>

<% int menuitem=1; // #### TEMPORARY, By Samson. If you want to modify, contact me! #### while (rset.next()) { %> <% if ( menuitem == 3) { %> <% }else{ %>
 • &area=hk"> <%=rset.getString("NAME")%>
<% } %> <% menuitem++;} %>

<% if (service_type.equals("B")) { %> <% } else {%> <% } %> <% String color="9ECDC3"; ResultSet rset4; String sql; // Connection of JDBC if (request.getParameter("area")==null) { sql = "select A.ID ID, C_NAME NAME, HOTLINE1, C_ADDRESS ADDRESS,to_char(POST_DATE,'YY.MM.DD') POST_DATE, to_char(OFF_DATE,'YY.MM.DD') OFF_DATE from dr_list A, dr_list_cat_type B where A.id = B.dr_list_id and A.status = 'C' and B.dr_list_cat_id =" + request.getParameter("service_id") + " and area='HK' and C_NAME is not null and OFF_DATE > sysdate "; } else { if (request.getParameter("area").equals("nt")) { sql = "select A.ID ID, C_NAME NAME, HOTLINE1, C_ADDRESS ADDRESS,to_char(POST_DATE,'YY.MM.DD') POST_DATE, to_char(OFF_DATE,'YY.MM.DD') OFF_DATE from dr_list A, dr_list_cat_type B where A.id = B.dr_list_id and A.status = 'C' and B.dr_list_cat_id =" + request.getParameter("service_id") + " and area='NT' and C_NAME is not null and OFF_DATE > sysdate "; } else if (request.getParameter("area").equals("kln")) { sql = "select A.ID ID, C_NAME NAME, HOTLINE1, C_ADDRESS ADDRESS,to_char(POST_DATE,'YY.MM.DD') POST_DATE, to_char(OFF_DATE,'YY.MM.DD') OFF_DATE from dr_list A, dr_list_cat_type B where A.id = B.dr_list_id and A
request.getParameter("service_id") + " and area='KLN' and C_NAME is not null and OFF_DATE > sysdate "; } else { sql = "select A.ID ID, C_NAM
ADDRESS,to_char(POST_DATE,'YY.MM.DD') POST_DATE, to_char(OFF_DATE,'YY.MM.DD') OFF_DATE from dr_list A, dr_list_cat_type B where A.id = B.dr_list_id and A
request.getParameter("service_id") + " and area='HK' and C_NAME is not null and OFF_DATE > sysdate "; }} if (service_type.equals("B")) { sql = sql + " order by POST_DATE D
= sql + " order by C_NAME"; } rset4 = stmt.executeQuery (sql); %> <% while (rset4.next()) { %> <% } %> <% color = color.equals("9ECDC3")?"D6F0DF":"9ECDC

> <% if (!(request.getParameter("area") == null)) { if (request.getParameter("area").equals("hk")) { out.print(" "); } if (request.getParameter("area").equals("nt")) { out.print(""); } if (request.getParameter("area").equals("kln")) { out.print(" "); } } %>
```

Google : DB SQL vulnerability

Query:

```
"Microsoft OLE DB Provider for ODBC Drivers error '80040e14'" filetype:asp
```



"Microsoft OLE DB Provider for ODBC Drivers error '80040e14'" filetype:asp - Google Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.google.ch/webhp?client=firefox-a&rls=org.mozilla:en-GB:official&channel=s&hl=en&source=hp&btnG=Google+Search#hl=en&client=firefox-a&channel=s&rls=org.mozill

Google
Most Visited
"Microsoft OLE DB Provider for ODB...
Web Images Videos Maps News Groups Gmail more ▾
Search settings | Sign in

"Microsoft OLE DB Provider for ODBC Drivers errc

Search: ● the web ○ pages from Switzerland

Web ➕ Show options... Results **1** - **10** of about **52,800** for **"Microsoft OLE DB Provider for ODBC Drivers error** '**80040e14**'**" filetype:asp**. (**0.28** seconds)

Technical Support for Microsoft OLE DB Provider for ODBC Drive...
Microsoft OLE DB Provider for ODBC Drivers error '80040e14' [Microsoft][ODBC Microsoft Access Driver] Syntax error (missing operator) in query expression **...**
www.comersus.org/displayMessage.asp?mid=37099 - Cached - Similar

Microsoft OLE DB Provider for ODBC Drivers error '80040e14 ...
DMXzone.com the place for all Web Designers and Developers using Dreamweaver! Here you can find tutorials, Dreamweaver extensions, plug-ins, **...**
www.dmxzone.com/replies.asp?NewsId=476 - Cached - Similar

ASP Microsoft OLE DB Provider for ODBC Drivers error
17 Dec 2003 **...** This is the error **Microsoft OLE DB Provider for ODBC Drivers error '80040e14'** [Microsoft][ODBC Microsoft Access Driver] Syntax error **...**
www.eggheadcafe.com/forums/ForumPost.asp?ID=7899... - Cached - Similar

inetserver asp db **Microsoft OLE DB Provider for ODBC Drivers error ...**
2 posts - Last post: 1 Aug 2006
Microsoft OLE DB Provider for ODBC Drivers error '80040e14' [Microsoft][ODBC Microsoft Access Driver] Syntax error in date in query **...**
www.eggheadcafe.com/aspnet_answers/.../post27557152.asp - Cached - Similar

Kattouf Internet Services Ltd.
This site may harm your computer.
Microsoft OLE DB Provider for ODBC Drivers error '80040e14' [Microsoft][ODBC Microsoft Access Driver] Syntax error in UPDATE statement. **...**
www.kattouf.com/faq.asp?categoryID=27&faqtype=2... - Similar

VP-ASP Shopping Cart - Online Help Guides - Other Database Errors
26 Feb 2007 **... Microsoft OLE DB Provider for ODBC Drivers error '80040e14'** [Microsoft][ODBC Microsoft Access 97 Driver] **...**
helpnotes.vpasp.com/shopexd.asp?id=777 - Cached - Similar

WEBULANT - Web and IT Outsourcing - Atlanta and Miami Web Design **...**
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'. ID: 96, Category: Database Support, Status: Resolved, Views: 2939. Question: **...**
www.webulant.com/customer_support/.../issue_view.asp?... - Cached - Similar

Done

Foot-printing: IP ranges

The question which IP numbers are available to the company is best answered by searching the database of RIR (Regional Internet Register). In Europe we query the link: http://www.ripe.net and for North America: http://www.arin.net .

For the query one needs only a single correct IP number pertinent to the company. A simple way to get it is via: nslookup fhnw.ch:

```
C:\Documents and Settings\ulisse>nslookup fhnw.ch

…

Non-authoritative answer:

Name:    fhnw.ch

Address:  147.86.3.160
```

Foot-printing: DNS query

Name server are responsible for the mapping of DN in IP addresses. A DNS query via dig tells more than that: it reveals names and IP addresses of important parts of the company’s infrastructure (i.e. the of the mail servers)

```
ulisse@beaver:~$ dig fhnw.ch

; <<>> DiG 9.7.0-P1 <<>> fhnw.ch

...

;; QUESTION SECTION:

;fhnw.ch.                    IN       A

;; ANSWER SECTION:

fhnw.ch.           28        IN       A        147.86.3.160

...

DNSSEC:N

Name servers:

ns.inwx.de

ns1.fhnw.ch        [147.86.3.20]

ns2.fhnw.ch        [147.86.3.21]
```

Foot-printing: automated DNS query

If we know the IP range for the company, we can easily do a reverse mapping in order to get the DN of the machines with a Perl script:

```
#! usr/bin/perl
use socket;
$b_net = “147.86.”;
for ($i=0; $i<255; $i++) {
  for ($j=0; $j<255; $j++) {
    $ipaddr = “$b_net.$i.$j”;
    $name = gethostbyaddr($ipaddr, AF_INET);
    if ($name) {
      print “${ip}\t${name}\n”;
   }}}
```

TCP or UDP proto

Scanning

Goal of the scanning phase:

1. Determine the network structure. This includes the search of all machines that are reachable from inside or outside the company.
2. Analyse the hosts in detail
3. Identify the OS and versions used
4. Determine the services running in the hosts and the corresponding software products. This can give hints at possible vulnerabilities present on a target system
5. Perform vulnerability scans.

Scanning: Network structure

`traceroute` is the tool of choice to analyse the network structure. Its main output is a list of all **hops** (IP addresses) till the target system.

There are different `traceroute` variants that either use `UDP` or `ICMP` packets:

On `*ix` systems, UDP is usually the default and ICMP can be used with the `-I` option. Its often a good idea to use both options, especially if one is not successful.

`traceroute` has many more options, the most important of them are:

a) `-n` Print hop addresses numerically rather than symbolically and numerically. This saves a NS address-to-name lookup for each gateway found on the path.

b) -d Set the `SO_DEBUG` socket option.

c) `-F` Set the "don't fragment" bit.

Scanning: traceroute (1)

Below the output of : `traceroute -q1 www.fhnw.ch`

```
ulisse@beaver:~$ traceroute -q 1 www.fhnw.ch
traceroute to www.fhnw.ch (147.86.3.160), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  1.952 ms
 2  zhhia00p-adsl15.bluewin.ch (85.3.128.1)  13.308 ms
 3  net1701.zhhia00p-rtdi02.bluewin.ch (213.3.247.190)  16.333 ms
 4  net1701.zhhdz09p-rtdi02.bluewin.ch (213.3.247.189)  18.048 ms
 5  198-0-186-195.bluewin.ch (195.186.0.198)  20.157 ms
 6  i79zhb-025-bun1.bb.ip-plus.net (138.187.129.113)  26.333 ms
 7  i79tix-025-ten1-1.bb.ip-plus.net (138.187.129.82)  24.471 ms
 8  swit-00-ser0.ce.ip-plus.net (164.128.22.130)  26.083 ms
 9  swiEZ2-10GE-1-3.switch.ch (130.59.36.249)  28.200 ms
10  swiBA2-10GE-1-4.switch.ch (130.59.37.106)  31.662 ms
11  unibi7-te-1-2.urz.p.unibas.ch (192.43.192.197)  30.541 ms
12  192.43.192.222 (192.43.192.222)  32.488 ms
13  * * * till hop 30
```

Router unib

Firewall

Scanning: `traceroute` to mail server

Output of: `traceroute  -n –q 1 mxmuu11.fhnw.ch`

```
traceroute to mxnmu11.fhnw.ch (147.86.3.24), 30 hops max, 60 byte packets
 1  192.168.1.1  4.601 ms  4.462 ms  4.336 ms
 2  85.3.128.1  12.362 ms  14.110 ms  16.254 ms
 3  213.3.247.190  14.538 ms  15.304 ms  16.790 ms
 4  213.3.247.189  33.385 ms  33.317 ms  33.226 ms
 5  195.186.0.198  33.103 ms  33.006 ms  32.944 ms
 6  138.187.129.113  32.771 ms  41.614 ms  40.523 ms
 7  138.187.129.82  40.179 ms  36.939 ms  35.468 ms
 8  164.128.22.130  29.013 ms  31.980 ms  34.119 ms
 9  130.59.36.249  33.521 ms  18.498 ms  20.060 ms
10  130.59.37.106  24.019 ms  26.989 ms  27.469 ms
11  192.43.192.197  40.458 ms  40.365 ms  30.570 ms
12  192.43.192.222  30.451 ms  20.893 ms  20.714 ms
13  * * *
14  * * * until hop 30
```

Scanning: traceroute from internal host

Output of: `traceroute  -n -q 1 www.fhnw.ch`

```
ulisse@beaver:~$ traceroute  -q 1 www.fhnw.ch
traceroute to www.fhnw.ch (147.86.3.160), 30 hops max, 60 byte packets
 1  10.212.136.1 (10.212.136.1)  1.287 ms
 2  nd41u101-sta-vl3213.net.fhnw.ch (10.212.16.33)  3.039 ms
 3  nc40u101-sta-vl3113.net.fhnw.ch (10.212.16.17)  1.516 ms
 4  nca0e001-sta-vl3113.net.fhnw.ch (10.218.0.17)  2.147 ms
 5  nda0e001-sta-vl3113.net.fhnw.ch (10.218.0.18)  1.986 ms
 6  *
 7  *
```

Scanning: `traceroute` from int. host to external

Output of: `traceroute –q 1 www.ethz.ch`

```
ulisse@beaver:~$ traceroute -q 1  www.ethz.ch
traceroute to www.ethz.ch (129.132.19.220), 30 hops max, 60 byte packets
 1  10.212.136.1 (10.212.136.1)  6.655 ms
 2  nd41u101-sta-vl3213.net.fhnw.ch (10.212.16.33)  2.251 ms
 3  nc40u101-sta-vl3113.net.fhnw.ch (10.212.16.17)  2.153 ms
 4  nca0e001-sta-vl3113.net.fhnw.ch (10.218.0.17)  2.618 ms
 5  nda0e001-sta-vl3113.net.fhnw.ch (10.218.0.18)  2.515 ms
 6  nfa0e002-sta.net.fhnw.ch (10.218.0.252)  2.402 ms
 7  nda0e001-sin-vl4064.net.fhnw.ch (193.73.125.14)  12.274 ms
 8  unibi7-vl-501.urz.p.unibas.ch (192.43.192.213)  3.674 ms
 9  swiba2.urz.p.unibas.ch (192.43.192.196)  5.537 ms
10  swiez2-10ge-5-4.switch.ch (130.59.37.105)  5.375 ms
11  rou-gw-rz-tengig-to-switch.ethz.ch (192.33.92.1)  5.281 ms
12  rou-fw-rz-rz-gw.ethz.ch (192.33.92.169)  4.453 ms
13  * * till hop 30
```

Scanning: hping3 (1)

hping3 allows to trace a route to a machine using UDP, TCP and ICMP.

Thus we can trace the route to using a TCP SYN port 80 probe.

```
ulisse@beaver:~$ hping3 --ttl 1  --traceroute --destport 80 --syn lis.technik.fhnw.ch
HPING lis.technik.fhnw.ch (eth0 147.86.20.21): S set, 40 headers + 0 data bytes
hop=1 TTL 0 during transit from ip=192.168.1.1 name=UNKNOWN
hop=1 hoprtt=0.6 ms
hop=2 TTL 0 during transit from ip=85.3.128.1 name=zhhia00p-adsl15.bluewin.ch
hop=2 hoprtt=13.0 ms
hop=3 TTL 0 during transit from ip=213.3.247.190 name=net1701.zhhia00p-rtdi02.bluewin.ch
hop=3 hoprtt=12.8 ms
hop=4 TTL 0 during transit from ip=213.3.247.189 name=net1701.zhhdz09p-rtdi02.bluewin.ch
hop=4 hoprtt=12.6 ms
hop=5 TTL 0 during transit from ip=195.186.0.198 name=198-0-186-195.bluewin.ch
hop=5 hoprtt=12.7 ms
hop=6 TTL 0 during transit from ip=138.187.129.113 name=i79zhb-025-bun1.bb.ip-plus.net
hop=6 hoprtt=15.1 ms
```

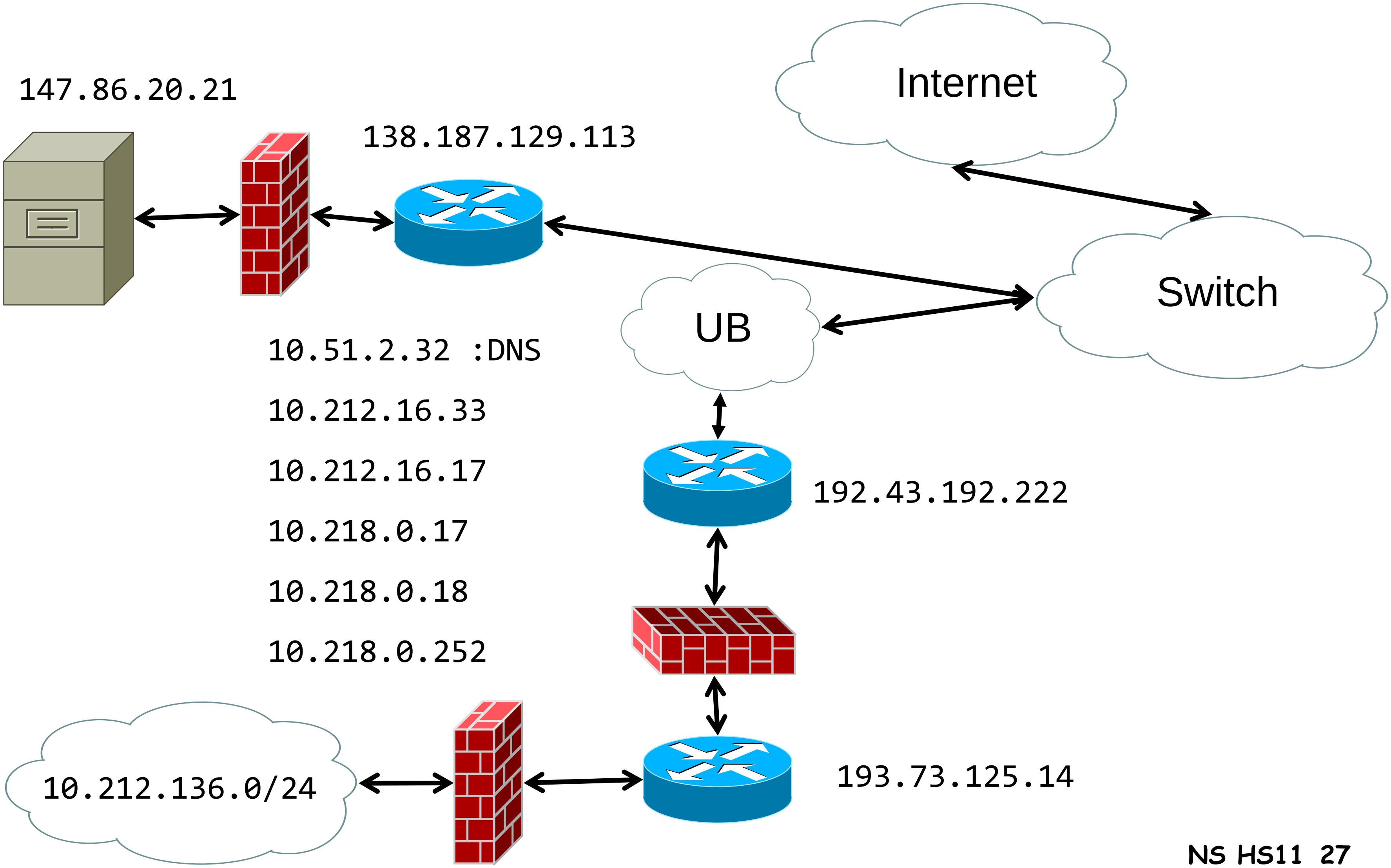
Scanning: hping3 (2)

hping3 option in the example:

1. --ttl Start with ttt set to 1 s
2. --traceroute Increment ttl for every subsequent attempt
3. --destport Set the destination port
4. --syn Set the SYN flag in TCP header.

The example shows that another router 138.187.129.113 is responsible for the route to lis.technik.fhnw.ch. Only 6 hops separate the external machine from lis.technik.fhnw.ch.

FHNW network so far



10.51.2.32 :DNS

10.212.16.33

10.212.16.17

10.218.0.17

10.218.0.18

10.218.0.252

nmap

nmap: network exploration tool

Purpose: “*Nmap ("Network Mapper") is a free and open source utility for network exploration or security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime*.”

Download: `http://www.insecure.org/nmap/`

Functionality: (1) Network mapping, (2) port scanning (3) OS detection and ping sweeps (and much more besides …)

nmap: functionality (1)

nmap uses the following scan techniques:

(1) UDP scan	(7) TCP FIN scan
(2) TCP connect()	(8) TCP ACK sweep
(3) TCP SYN (half open)	(9) Xmas Tree (FIN, URG, PSH flags set)
(4) ftp proxy (bounce attack)	(10) TCP SYN sweep
(5) Reverse-Identification	(11) IP Protocol
(6) ICMP (ping sweep)	(12) Null Scan (FIN, URG, PSH, RST, ACK,SYN flags not set)

nmap: functionality (2)

nmap uses the following OS detection techniques:

→ TCP/IP fingerprinting
→ stealth scanning
→ dynamic delay and retransmission calculations
→ parallel scanning (-Pn)
→ detection of down hosts via parallel pings
→ decoy scanning
→ port filtering detection
→ direct (non-port mapper) RPC scanning
→ fragmentation scanning
→ flexible target and port specification.

nmap: Labor Bedingungen

1. Die `nmap` Scan-Rate auf ca 10/s reduzieren;

2. Ports Scan nur an maximal 3 FHNW-Server durchführen;

3. Bevor Sie zu scannen beginnen, schreiben Sie ein Email an network.services@fhnw.ch mit folgenden Inhalt:
   a) Aktuelle IP-Nummer Ihres Laptops;
   b) IP-Nummer Bereich, den Sie scannen wollen;
   c) Zeitfenster des Scans (ungefähr von ... bis ...)

Scanning: network's mapping

Goal # 1: Network Mapping

Why: To determine the topology of the network.

How:

1. Manually using tools like `ping`, `traceroute`, (Windows: `tracert`)
2. Automatically with tools like Google based TouchGraph network mapping tool
3. Semi-automatically with `nmap` $\geq$ 5.5 (Zenmap GUI tool)

Scanning: ports open/closed (1)

Goal # 2: **Port Scanning**

Why: To find open ports in order to exploit them.

How: With `nmap`.

- **TCP Connect**: attempt to complete the 3-way handshake, look for SYN-ACK. This scan is easy to detect.
- **TCP SYN Scan**: “half-open” scan, look for SYN-ACK, then send RESET, in this case the target system will not record the attempted connection. It is faster than the TCP connect scan.
- **TCP FIN, Xmas Tree, Null Scans**: scans that violate the protocol: the closed ports send RESET, the open ones send nothing (Windows does not respond to these scans).

Scanning: ports open/closed (2)

- **TCP ACK Scan**: may be useful to get past packet filters (believes it is a response to a request from inside a firewall), if one receives a RESET, one knows that this port is open through the firewall.
- **FTP Bounce Scan**: request a server to send a file to a victim machine inside its network (most servers though, have this service disabled).
- **UDP Scan**: if receive ICMP Port Unreachable, it assumes that port closed, otherwise open. (Unreliable).
- **Ping Sweep**: can use ICMP or TCP packets to identify active hosts within the target network.

Scanning: ports open/closed (3)

Additional goals of a network scan:

- → **Decoys:** insert false IP addresses in scan packets.
- → **Ping Sweeps**: identify active hosts on a target network.
- → **Find RPCs**: connect to each open port looking for common RPC services (send a NULL RPC commands).

Scanning: OS detection (1)

Goal # 3: Operating System Detection

Why: To determine which Operating System is in use in order to exploit known vulnerabilities.

Also known as *TCP stack fingerprinting*. It takes advantage of the ambiguity of how to handle illegal combinations of TCP code bits that is found in the RFCs.

Experience teaches that each OS responds to illegal combinations in different ways. Therefore one determines the OS by examining the system's responses.

Scanning: OS detection (2)

How: With `nmap` you examine how the OS manipulates specific parameters of a TCP-packet.

→ **Window Size**: most *ix OS keep the same window size throughout a session. Windows OS tend to change the window size during a session.

→ **Time to Live (TTL)**: FreeBSD or Linux typically use 64; Windows OS typically uses 128.

→ **Do Not Fragment Flag**: most OS leave this flag set, OpenBSD leaves it unset.

Scanning: vulnerability assessment

Goal # 4: Vulnerability Assessment

Why: To determine what known (or unknown?) vulnerabilities exist on a given network

Vulnerabilities come from:

→ Default configuration weakness

→ Configuration errors

→ Security holes in specific versions of applications and protocols

→ Failure to download security patches!

Scanning: vulnerability checkers

Vulnerability checkers consists of:

→ Database of known vulnerabilities

→ Configuration tool

→ Scanning engine

→ Knowledge base of current scan

→ Report generation tool

Vulnerability checker tool: Nessus

Purpose: “*A software which will audit remotely a given network and determine whether bad guys (aka 'crackers') may break into it, or misuse it in some way*.”

Available platforms: UNIX for client and server
Windows for client only

Available at: **`http://www.nessus.org/`**

What does Nessus do?

(1) It iteratively tests if a target system (or systems) is vulnerable to **known** exploits.

(2) It uses a separate plug-in (written in C or in Nessus scripting language) for each security test the user wants to execute.

(3) It can test multiple hosts concurrently.

(4) It produces a thorough vulnerability assessment report at the conclusion of the vulnerability scan.

What does Nessus check for?

→ Backdoors

→ CGI abuses

→ Denial of Service

→ Finger abuses

→ FTP

→ Gain a shell remotely

→ Gain root remotely

→ Port scanners

→ Remote file access

→ RPC

→ SMTP problems

→ Useless services

→ Windows loopholes

→ and more...

Traffic shaping and Intrusion Detection Systems (IDS)

Traffic shaping

Traditional firewall is a binary system:

→ Allow traffic or disallow traffic

Traffic shaping is a more subtle technique:

→ it limits certain kinds of traffic;

→ it can differentiate by host address, by protocol, etc

→ Multi-Protocol Label Switching (MPLS):

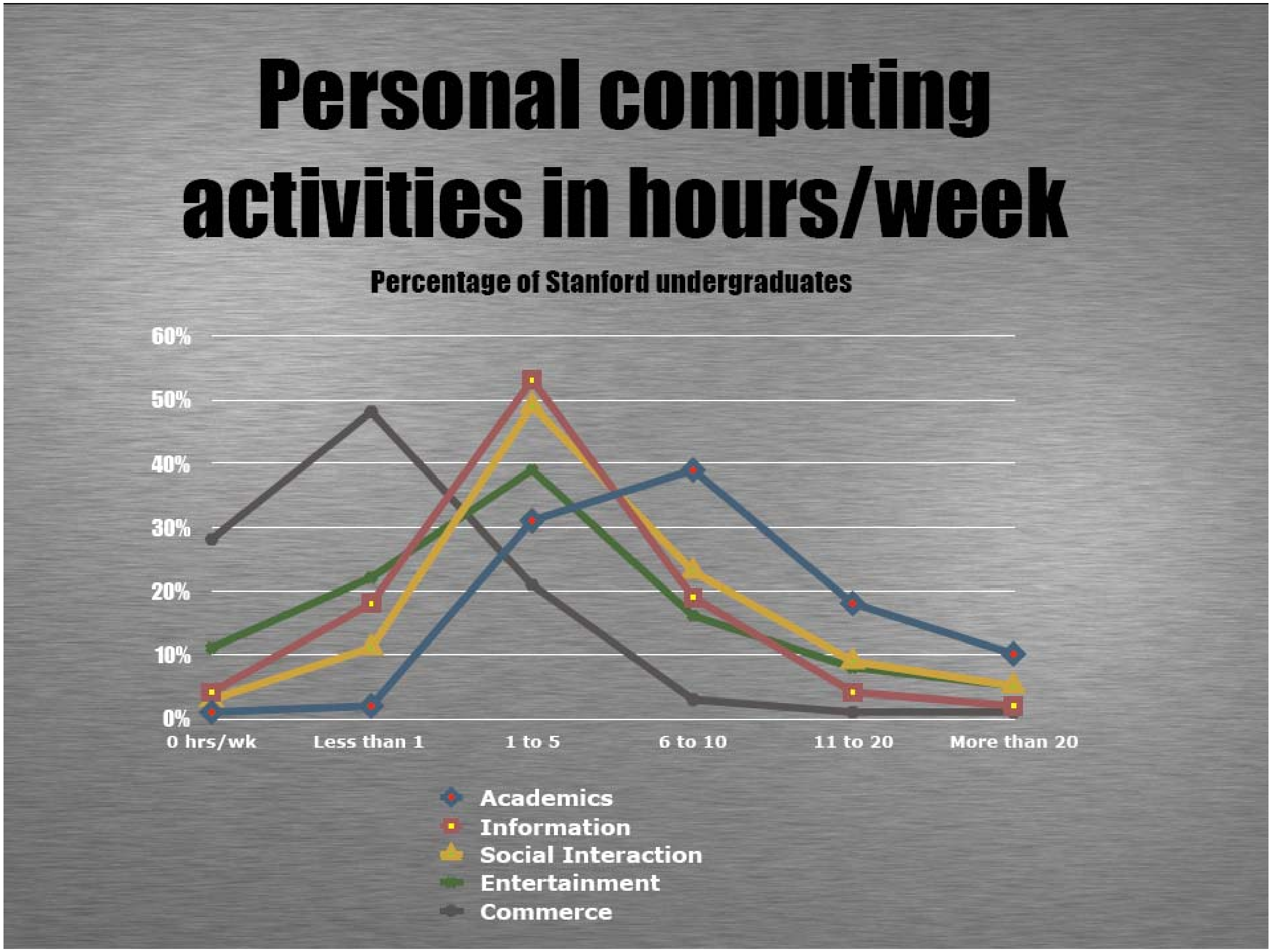
Label traffic flows at the edge of the network and let core routers identify the required class of service

With traffic shaping one can solve a fastidious problem known on every school's campus:

→ `P2P` file sharing takes a lot of bandwidth

→ On average 1/3 of a university's network bandwidth is consumed by `BitTorrent` (you know what I mean …)

Academic computer users' patterns



Traffic shaping functions

1. Classify and analyze traffic:

   Classify by IP address and port number

   Use application-specific information (layer 7)

2. Control traffic:

   Selectively slow certain classes of traffic

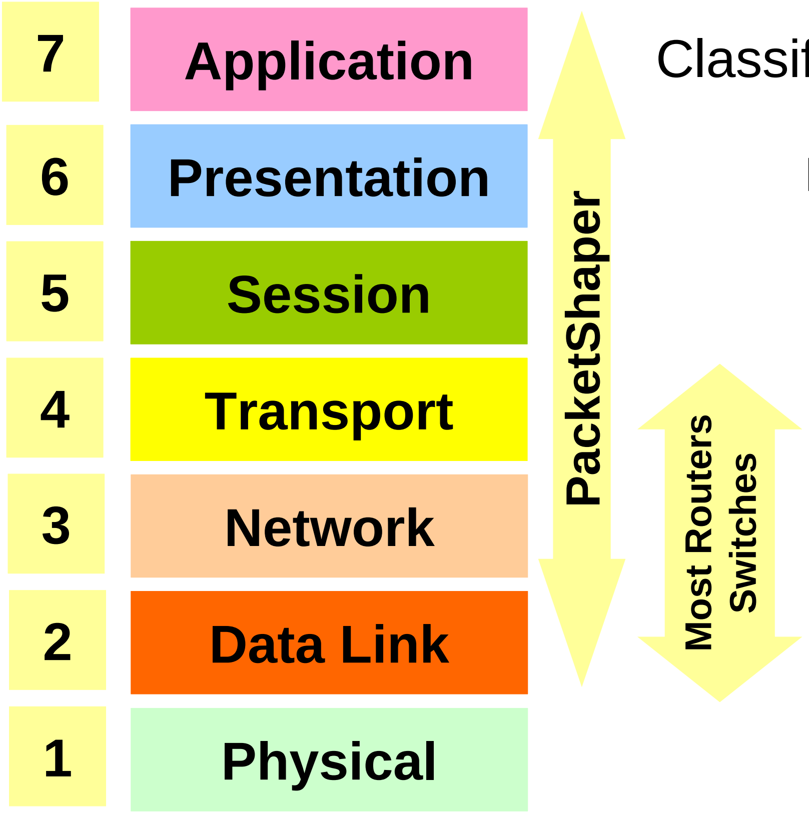
3. Monitor network performance:

   Collect performance data, used to improve policies

4. Network resilience:

   Active traffic management can provide resilience to DoS attacks, at least within the enterprise network

PacketShaper classification



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

PacketShaper

Most Routers
Switches

Classify more than 400 apps at OSI Layers 2-7

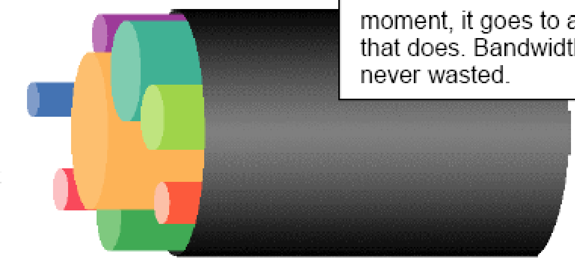
Peer-to-Peer Apps:

- Aimster
- AudioGalaxy
- CuteMX
- DirectConnect
- Gnutella
- Hotline
- iMesh
- KaZaA/Morpheus
- Napster
- ScourExchange
- Tripnosis….

Some Other Apps:

- H.323
- RTP-I/RTCP-I
- PASV FTP
- HTTP
- Real
- WinMedia
- Shoutcast
- MPEG
- Quicktime
- RTSP
- Chatting Apps
- Games

PacketShaper controls



It's as if each application or type of traffic gets its own appropriately sized link.

If an application doesn't need its bandwidth at the moment, it goes to another that does. Bandwidth is never wasted.

A partition:

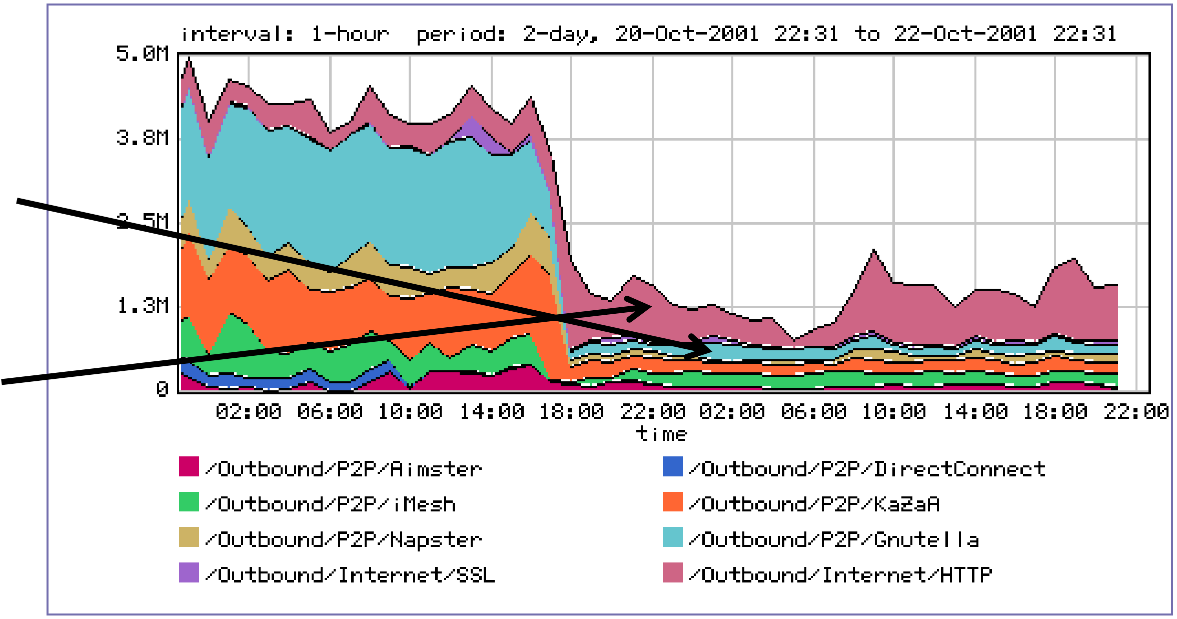
→ Creates a virtual pipe within a link for `each traffic class;

→ Provides a minimal and maximal bandwidth for each class;

→ So it enables an efficient bandwidth's use

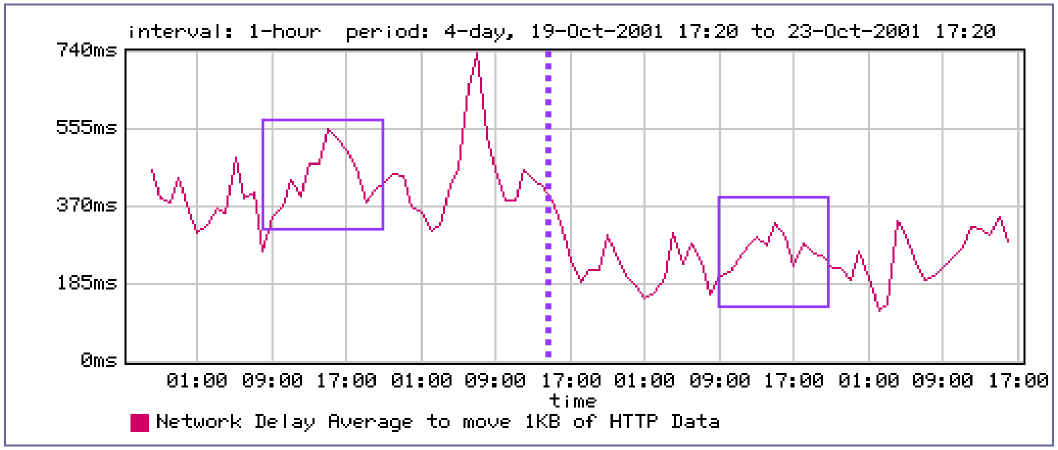
Rate shaped P2P capped at 300kbps

Rate shaped HTTP/SSL to give better performance



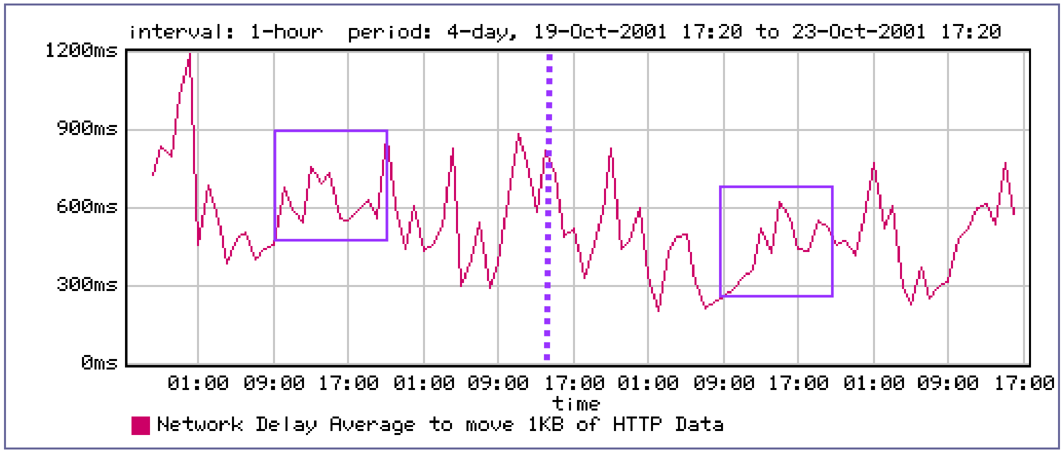
PacketShaper report: HTTP

Outside Web Server Normalized
Network Response Times



No Shaping **Shaping**

Inside Web Server Normalized
Network Response Times



No Shaping **Shaping**

Host and network intrusion detection

Intrusion prevention:

1. Network firewall:
   → Restrict flow of packets (see firewall slides);
2. System security:
   → Find buffer overflow vulnerabilities and remove them!

Intrusion detection:

1. Discover system modifications:
   → Tripwire
2. Look for attack in progress:
   → Network traffic patterns
   → System calls, other system events

Tripwire

Standard modus operandi of a cracker's attack:

a) Gain user access to system;
b) Gain root access;
c) Replace system binaries to set up backdoor;
d) Use backdoor for future activities.

Tripwire detects an attack by examining the system's binaries:

→ It computes hash of key system binaries;
→ it compares the actual hash to the hash it stored earlier;
→ It reports a problem if the hash is different;
→ It stores the reference hash codes on a read-only medium.

How to outsmart Tripwire

Cracker's attack with a new twist:

1. Gain access;
2. Install backdoor:

   This can be stored in main memory, not on disk!!
3. Use it.

What can Tripwire do in this case?

- → Not much because this attack doesn't change the system files stored on the hard disk!
- → Nevertheless using Tripwire is always a good idea.
- → It detects a compromised system's file after the attack has happened.

How to detect modified binary in memory

Can use system-call monitoring techniques. For example (see Wagner, Dean; *IEEE Security and Privacy Conf*. 2001):

→ Build automaton of expected (typical) system calls

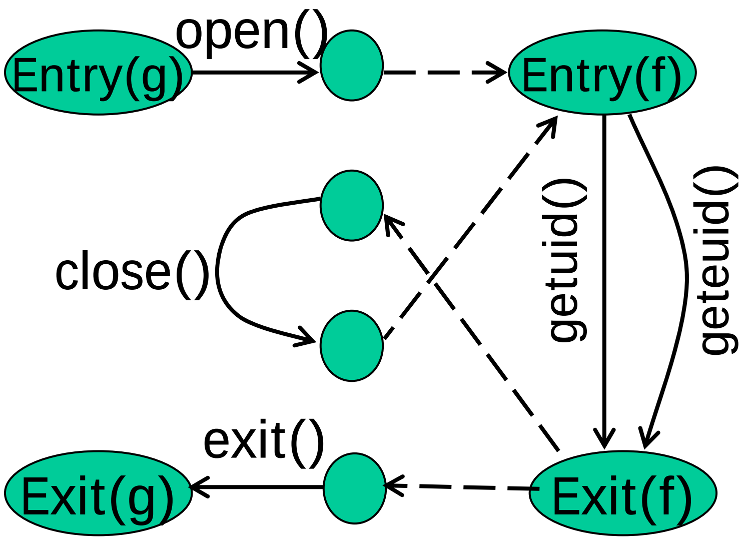
  – Can be done automatically from source code.

→ Monitor system calls from each program;

→ Compare with automaton and eventually catch violation.

Code's example and its relevant automaton

```
f(int x) {
   x ? getuid() : geteuid();
   x++
}
g() {
   fd = open("foo", O_RDONLY);
   f(0); close(fd); f(1);
   exit(0);
}
```



If code behavior is inconsistent with the automaton then something is wrong.

General intrusion detection

Many intrusion detection systems that are available, are roughly divided in three categories: (i) Network-based, (ii) host-based, or (iii) a combination of (i) and (ii).

Two basic models:

→ **Misuse** detection model:

- Maintain data on known attacks;
- Look for activity with corresponding signatures.

→ **Anomaly** detection model:

- Try to figure out what is “normal” (hard) and then
- Report anomalous behavior

Fundamental problem: too many **false** alarms.

`http://www.snort.org/`

Example of misuse that leads to detection: `rootkit`

A typical `rootkit` sniffs networks for passwords:

- It is a collection of programs that allow attacker to install and operate a packet sniffer (on Unix machines).

`rootkit` attack:

- Use stolen password or dictionary attack to get access as a legitimate user;
- Get `root` access using vulnerabilities in `rdist, sendmail, /bin/mail, loadmodule, rpc.ypupdated (NIS` Network Information Service data base), `lpr`, or `passwd.`
- Via `ftp` the `rootkit` is uploaded to the host, unpacked, compiled, and installed.
- It then collects more username/password pairs and then moves on.

Rootkit covers its tracks

Modifies netstat, ps, ls, du, ifconfig, login

- Modified binaries hide the new files used by **rootkit**
- Modified login allows attacker to return for fishing new passwords

Rootkit fools simple Tripwire checksum

- Modified binaries have the same checksum as the correct one.
- But a better hash than MD5 would make a rootkit attack more difficult (e.g. SHA-1024).

Detecting `rootkit` on system

Not the best way:

- Disk is full of sniffer logs.

Manual confirmation:

- Reinstall a clean version of `ps` and see what processes are running.

Automatic detection:

- `rootkit` does not alter the data structures normally used by `netstat, ps, ls, du, ifconfig` only their output is faked;
- Thus a host-based intrusion detection can find `rootkit` files, as long as an update version of `rootkit` does not disable your intrusion detection system …

Detecting network attack (Sept. 2003)

Symantec *honeypot* running Red Hat Linux 9.

Attack

- Samba ‘`call_trans2open`’ Remote Buffer Overflow (BID 7294)
- Attacker installed a copy of the SHV4 `Rootkit`

Snort NIDS generated alerts against this attack from its standard rule signature:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 \
msg:"NETBIOS SMB trans2open buffer overflow attempt"; \
flow:to_server,established; \
content:"|00|"; offset:0; depth:1; \
content:"|ff|SMB|32|"; offset:4; depth:5; \
content:"|0014|"; offset:60; depth:2; \
     ...
```

More info: Symantec030929-Analysis-SHV4Rootkit.pdf
on the course's webpage.

Misuse's example: port sweep

Attacks can be OS specific:

- Bugs in specific OS implementations can be exploited to mount an attack;
- Oversights in default configuration's files open a path that can be easily exploited.

Attacker sweeps network to find vulnerabilities:

- Port sweep tries many ports on many IP addresses
- If the characteristic behavior is detected, then it mounts an attack
  - SGI IRIX responds to TCPMUX port (TCP on port 1)
  - If a machine responds, then SGI IRIX vulnerabilities can be tested and used to break in

Port sweep activity is easily detected.

Anomaly Detection

Basic idea:

- → Monitor network traffic, system calls;
- → Compute statistical properties;
- → Report errors if statistics lies outside an empirical established range.

Example: IDES (Denning, SRI)

- → For each user, store the daily count of certain activities
  - E.g., Fraction of hours spent reading email.
- → Maintain a list of counts for several days;
- → Report anomaly if count is outside weighted norm.

The crux is that the most unpredictable user is the most dangerous.

Anomaly: sys calls' sequences

Build traces during normal run of program:

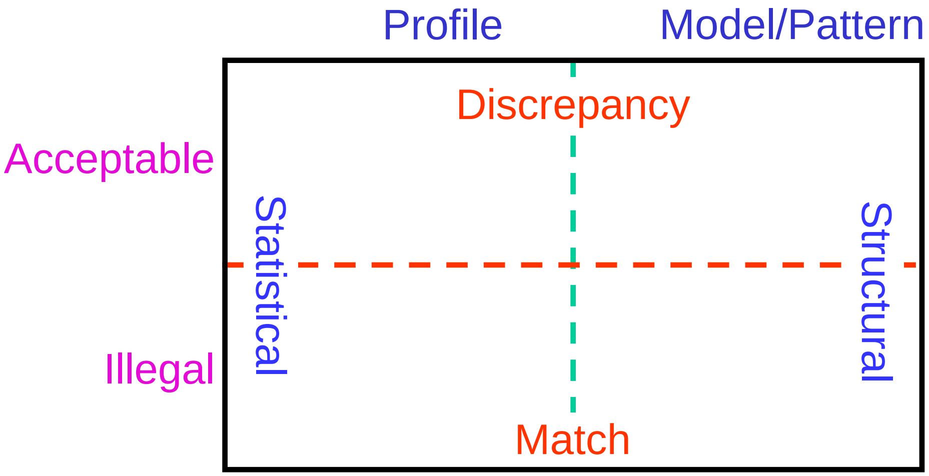
- Example of program's (good) behavior (sys calls), `open read write open mmap write fchmod close`
- Sample traces are stored in file (as 4-calls sequences):

```
open read write open
read write open mmap
write open mmap write
open mmap write fchmod
mmap write fchmod close
```

- Report anomaly if for example the following sequence is observed:

```
open read read open mmap write fchmod
close
```

Compute # of mismatches to get mismatch rate.



(See papers of Hofmeyr, Somayaji, Forrest)

Difficulties in intrusion detection

Lack of training data:

- → Lots of “normal” network and system call data but too little data containing realistic attacks, anomalies

Data drift:

- → Statistical methods are used to detect changes in behavior
- → That means that the cracker can attack gradually and incrementally thus defeating a statistical analysis.

Main characteristics not well understood:

- → By many measures, attack may be within bounds of “normal” range of activities

False positive are very costly:

- → System administrators spend many hours examining bogus evidence.

Example: strategic intrusion assessment (Lunt 1999)

Test over two-week period:

→ AFIWC's (US Air Force Information Warfare Center) intrusion detectors at 100 AFBs alarmed on 2 million sessions

→ Manual review identified **12,000** suspicious events

→ Further manual review ⇒ **four** actual incidents

Conclusion:

→ Most alarms are **false positives**;

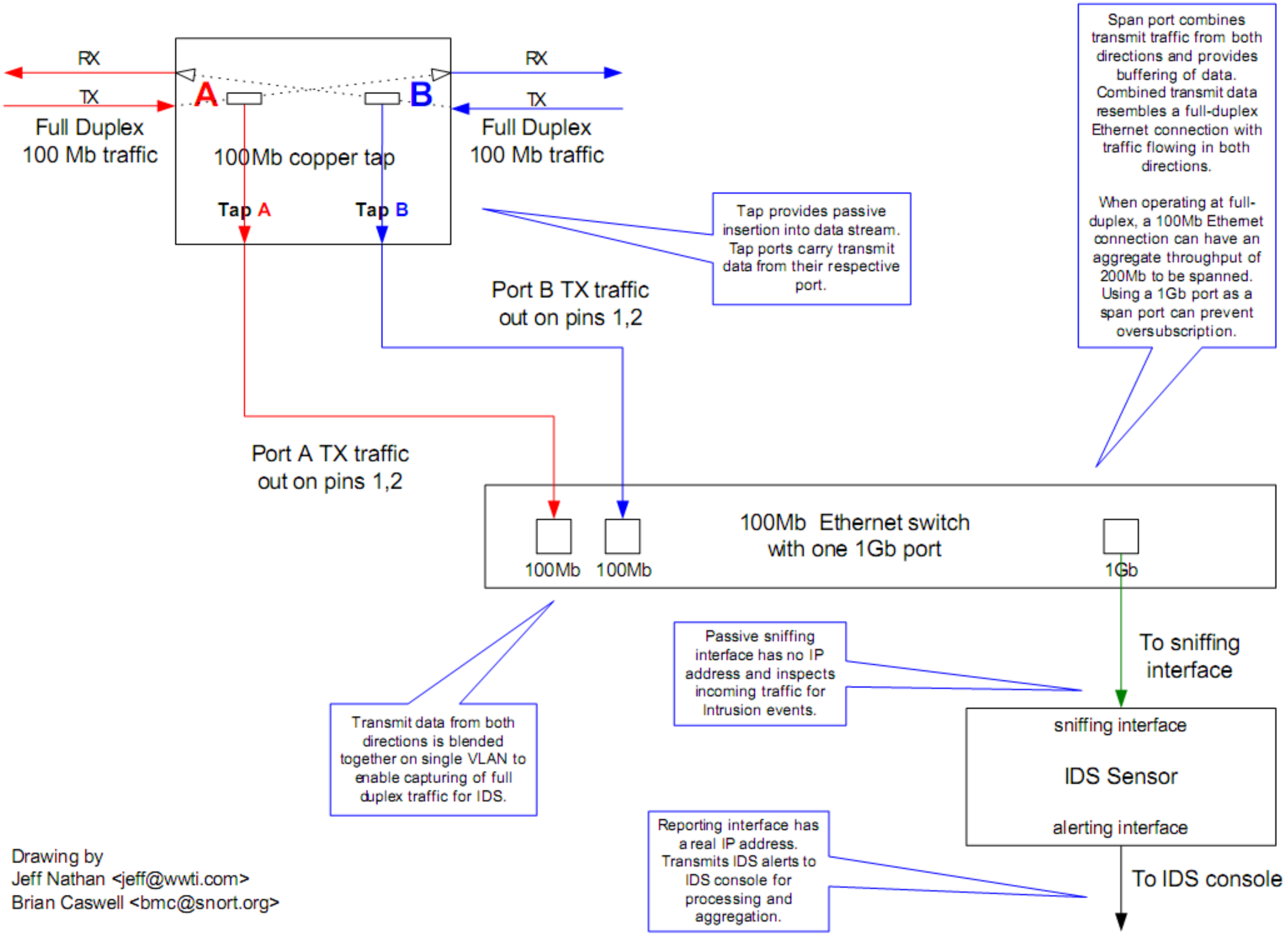
→ Most **true positives** are **trivial** incidents;

→ Of the significant incidents, most are isolated attacks to be dealt with **locally**.

See details in: `www.blackhat.com/presentations/bh-usa-99/teresa-lunt/tutorial.ppt`

Appendices

Appendix A: How to attach a IDS in a LAN



Tap provides passive insertion into data stream. Tap ports carry transmit data from their respective port.

Span port combines transmit traffic from both directions and provides buffering of data. Combined transmit data resembles a full-duplex Ethernet connection with traffic flowing in both directions.

When operating at full-duplex, a 100Mb Ethernet connection can have an aggregate throughput of 200Mb to be spanned. Using a 1Gb port as a span port can prevent oversubscription.

Transmit data from both directions is blended together on single VLAN to enable capturing of full duplex traffic for IDS.

Passive sniffing interface has no IP address and inspects incoming traffic for Intrusion events.

Reporting interface has a real IP address. Transmits IDS alerts to IDS console for processing and aggregation.

Drawing by
Jeff Nathan <jeff@wwti.com>
Brian Caswell <bmc@snort.org>

Appendix B: Packet analysis (1)

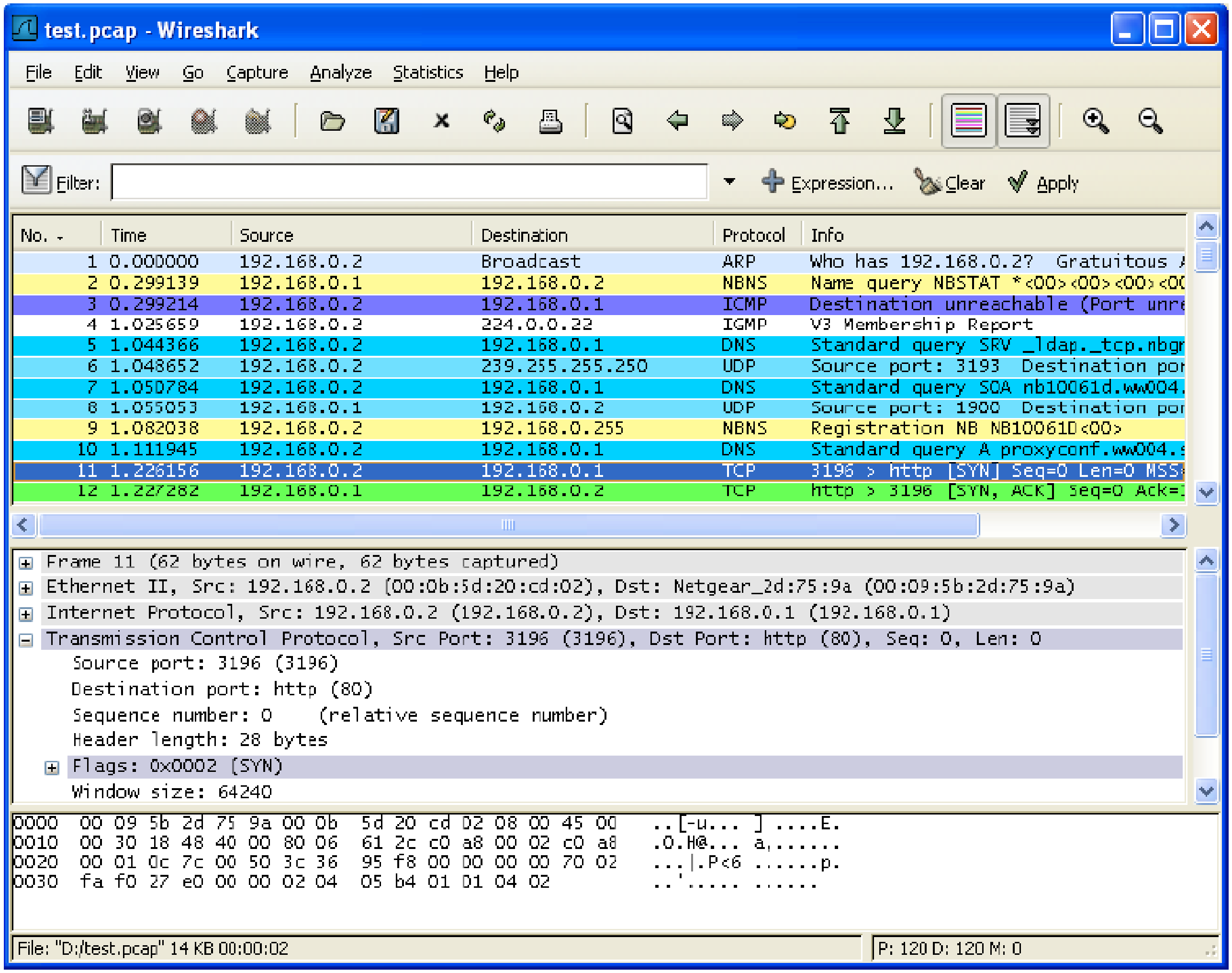
Goal: Capture and decode the header and body information used in different Internet protocols.

Why: The careful study of the packet flow in a network permits (i) to understand whether all components work as expected and (ii) to detect anomalies in the traffic due to an attack on the system.

How:

- Wireshark for both Unices and Windows: `www.wireshark.org`
- On Linux: `tcdump`

Appendix B: Packet analysis (2)



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.2	Broadcast	ARP	Who has 192.168.0.2? Gratuitous A
2	0.299139	192.168.0.1	192.168.0.2	NBNS	Name query NBSTAT *<00><00><00><00
3	0.299214	192.168.0.2	192.168.0.1	ICMP	Destination unreachable (Port unre
4	1.025659	192.168.0.2	224.0.0.22	IGMP	V3 Membership Report
5	1.044366	192.168.0.2	192.168.0.1	DNS	Standard query SRV _ldap._tcp.nbgr
6	1.048652	192.168.0.2	239.255.255.250	UDP	Source port: 3193 Destination por
7	1.050784	192.168.0.2	192.168.0.1	DNS	Standard query SOA nb10061d.ww004.
8	1.055053	192.168.0.1	192.168.0.2	UDP	Source port: 1900 Destination por
9	1.082038	192.168.0.2	192.168.0.255	NBNS	Registration NB NB10061D<00>
10	1.111945	192.168.0.2	192.168.0.1	DNS	Standard query A proxyconf.ww004.s
11	1.226156	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0 Len=0 MSS=
12	1.227282	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] Seq=0 Ack=1

```
⊞ Frame 11 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02), Dst: Netgear_2d:75:9a (00:09:5b:2d:75:9a)
⊞ Internet Protocol, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
⊟ Transmission Control Protocol, Src Port: 3196 (3196), Dst Port: http (80), Seq: 0, Len: 0
    Source port: 3196 (3196)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x0002 (SYN)
    Window size: 64240
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] .....E.
0010  00 30 18 48 40 00 80 06  61 2c c0 a8 00 02 c0 a8   .0.H@... a,......
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00 00 00 70 02   ...|.P<6 ......p.
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01 04 02         ..'..... ......
```

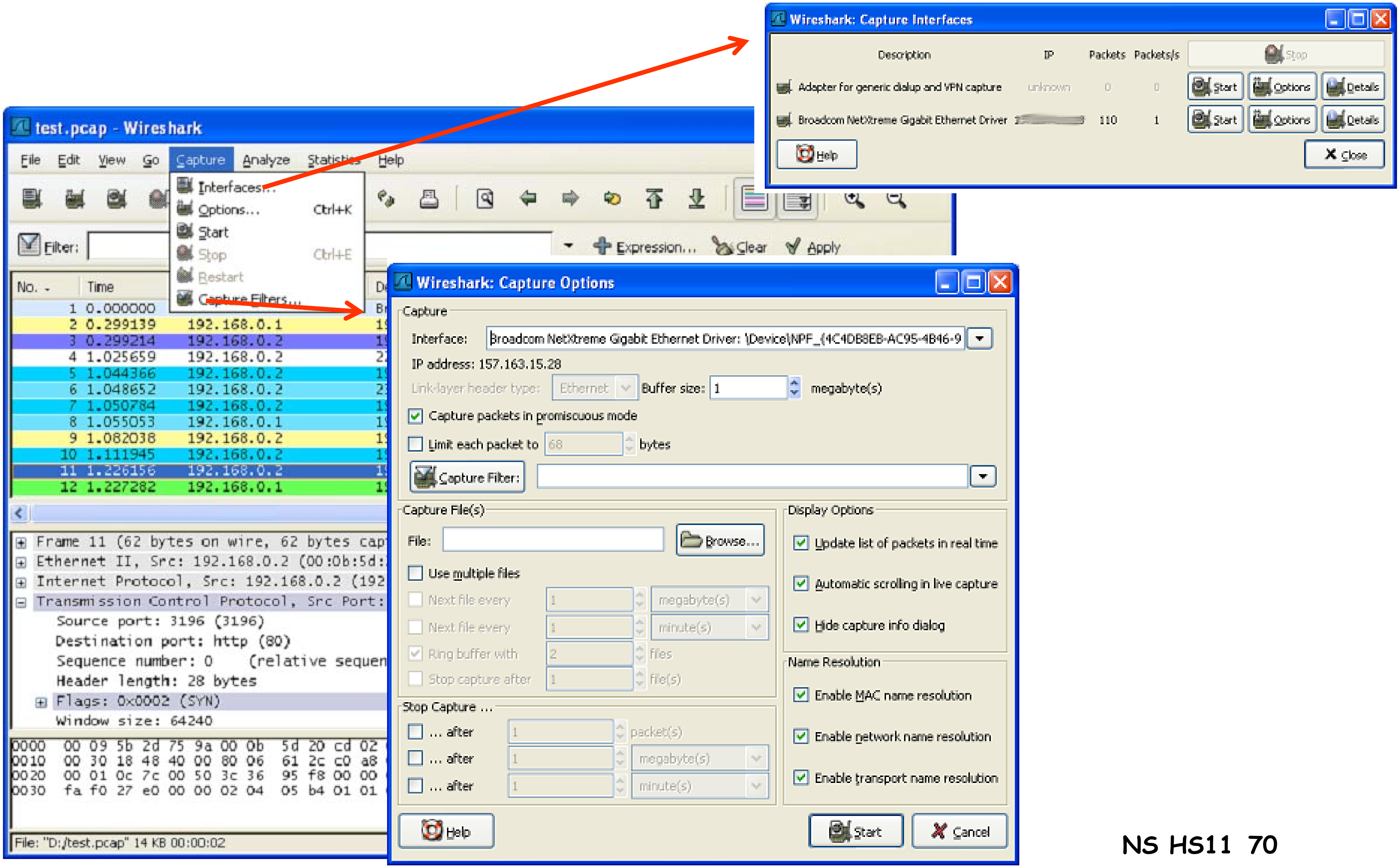
File: "D:/test.pcap" 14 KB 00:00:02 | P: 120 D: 120 M: 0

The packet list pane displays a summary of each packet captured.

The packet details pane displays the packet selected in the list pane with more details.

The packet bytes pane displays the ASCII data from the selection in the first pane.

Appendix B: Packet analysis (3)



test.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Capture menu: Interfaces... | Options... Ctrl+K | Start | Stop Ctrl+E | Restart | Capture Filters...

Filter: | Expression... | Clear | Apply

No.	Time	
1	0.000000	
2	0.299139	192.168.0.1
3	0.299214	192.168.0.2
4	1.025659	192.168.0.2
5	1.044366	192.168.0.2
6	1.048652	192.168.0.2
7	1.050784	192.168.0.2
8	1.055053	192.168.0.1
9	1.082038	192.168.0.2
10	1.111945	192.168.0.2
11	1.226156	192.168.0.2
12	1.227282	192.168.0.1

```
⊞ Frame 11 (62 bytes on wire, 62 bytes cap
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:
⊞ Internet Protocol, Src: 192.168.0.2 (192
⊟ Transmission Control Protocol, Src Port:
    Source port: 3196 (3196)
    Destination port: http (80)
    Sequence number: 0    (relative sequen
    Header length: 28 bytes
  ⊞ Flags: 0x0002 (SYN)
    Window size: 64240
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02
0010  00 30 18 48 40 00 80 06  61 2c c0 a8
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01
```

File: "D:/test.pcap" 14 KB 00:00:02

Wireshark: Capture Interfaces

Description	IP	Packets	Packets/s	Stop
Adapter for generic dialup and VPN capture	unknown	0	0	Start Options Details
Broadcom NetXtreme Gigabit Ethernet Driver	[redacted]	110	1	Start Options Details

Help | Close

Wireshark: Capture Options

Capture

Interface: Broadcom NetXtreme Gigabit Ethernet Driver: \Device\NPF_{4C4DB8EB-AC95-4B46-9

IP address: 157.163.15.28

Link-layer header type: Ethernet Buffer size: 1 megabyte(s)

☑ Capture packets in promiscuous mode

☐ Limit each packet to 68 bytes

Capture Filter:

Capture File(s)

File: | Browse...

☐ Use multiple files

☐ Next file every 1 megabyte(s)

☐ Next file every 1 minute(s)

☑ Ring buffer with 2 files

☐ Stop capture after 1 file(s)

Stop Capture ...

☐ ... after 1 packet(s)

☐ ... after 1 megabyte(s)

☐ ... after 1 minute(s)

Display Options

☑ Update list of packets in real time

☑ Automatic scrolling in live capture

☑ Hide capture info dialog

Name Resolution

☑ Enable MAC name resolution

☑ Enable network name resolution

☑ Enable transport name resolution

Help | Start | Cancel