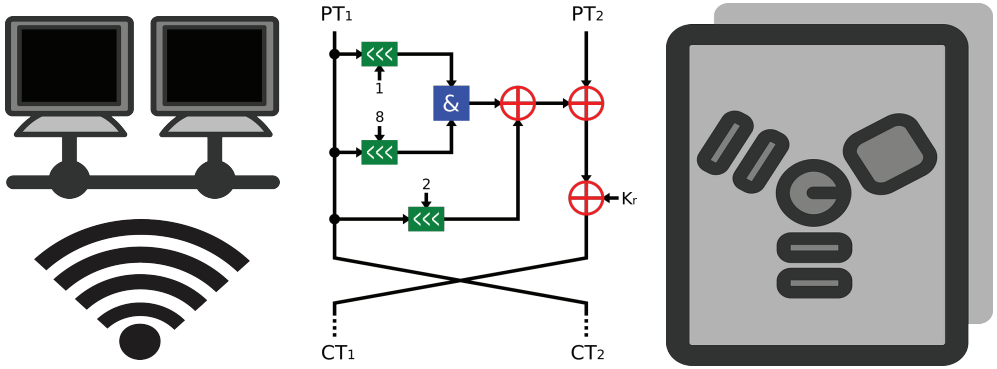
CSE 40567 / 60567: Computer Security



Network Security 5

Homework #7 is Due on 4/28
at 11:59PM (your timezone)

See **Assignments Page** on the course website for details

Final Exam Plan:

- Final will be released at 2pm on 5/7
- You will have 24hrs to complete it (due 2pm on 5/8)
- Open book / notes / Internet
- Format is short answer
- See the topic checklist on the course website

Course Instructor Feedback (CIF)

Deadline: 11:59PM, 5/3/20

***Being used this semester to assess online learning

Guest Lecture Thurs. 4/23: Stephen Watt on hacking, prison, and what came after

Live on zoom!

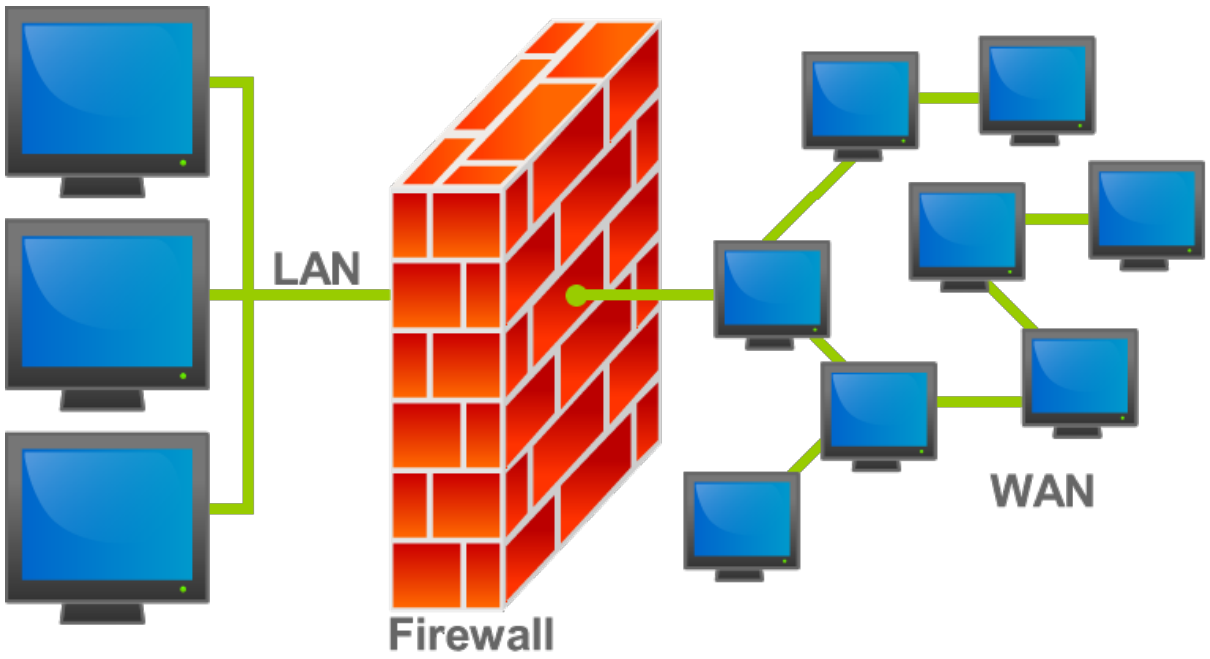
Firewalls

Firewalls

- Monitor and control all incoming and outgoing packets
- Firewall behavior is defined by a ruleset
- Two different categories
  - Network-based
  - Host-based

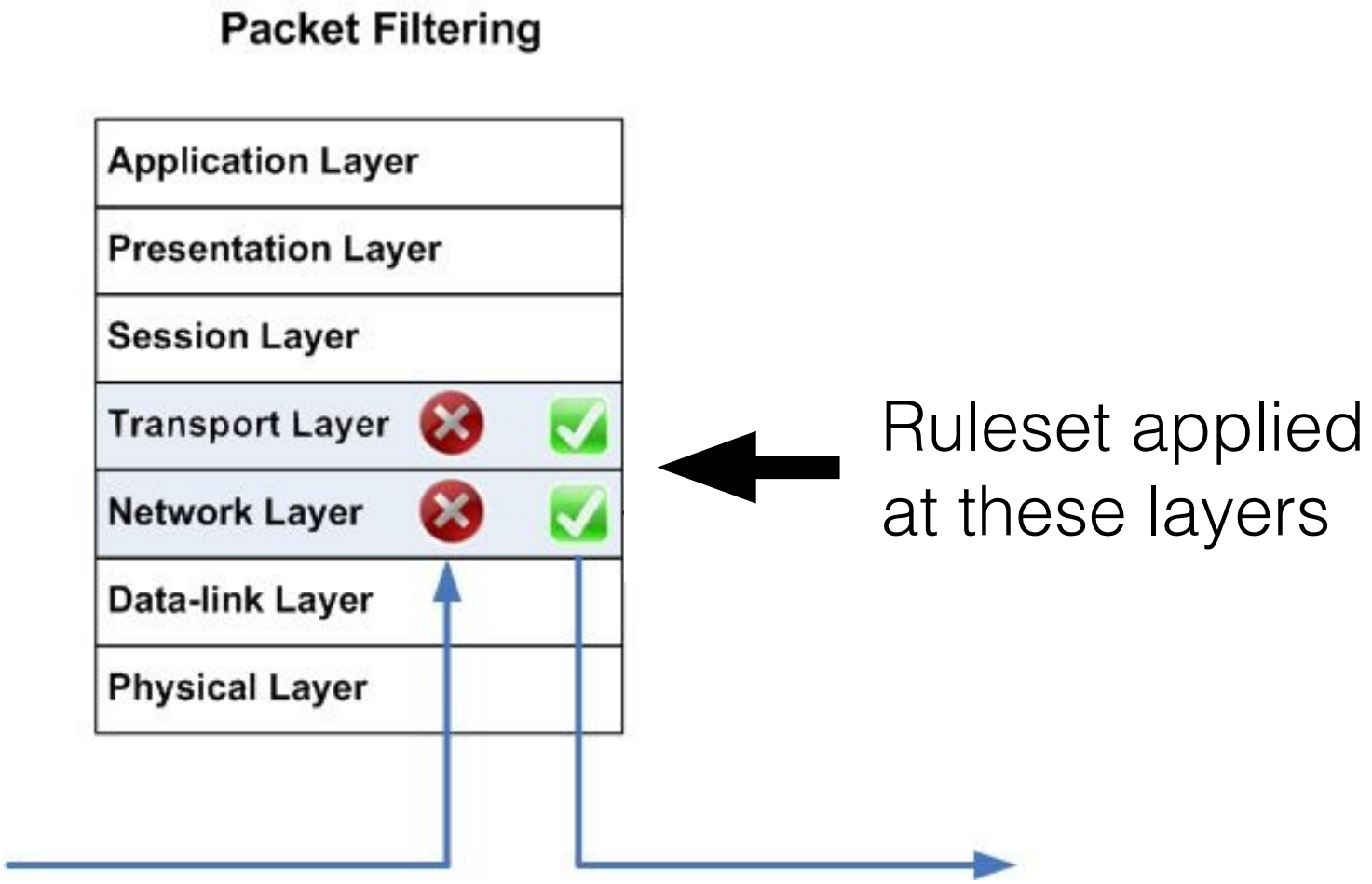
Firewall-X400

Firewall placement



An illustration of where a firewall would be located in a network.

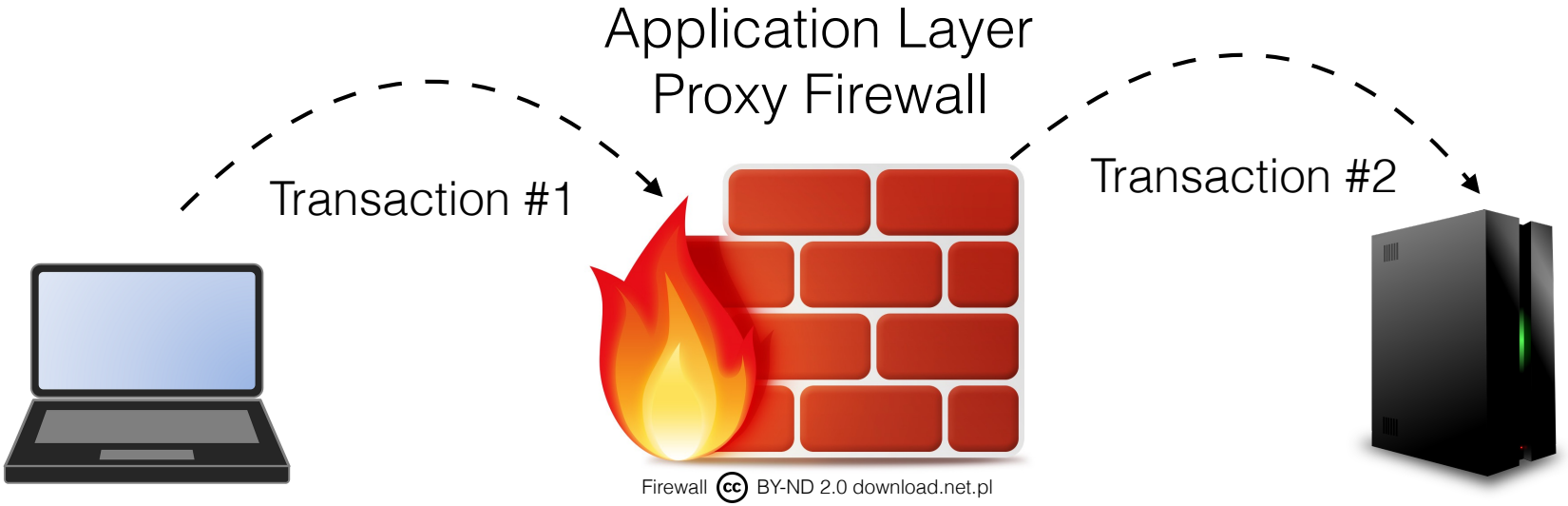
Network Layer / Packet Filtering



OSI Packet Filter CC BY-SA 3.0 Willysaef

Application Layer

Split transaction in two:

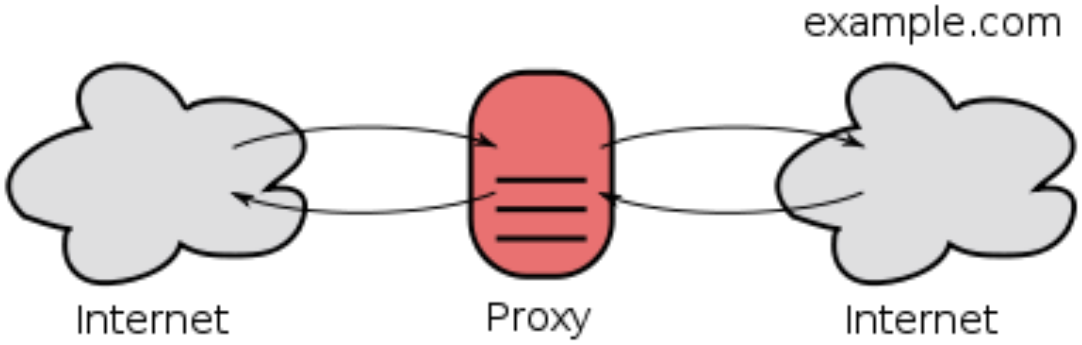


To the client, the firewall appears to be the server

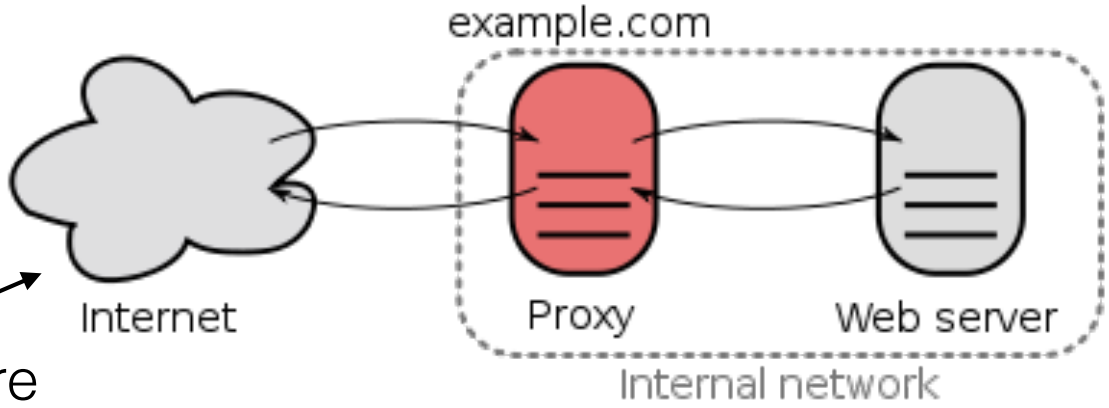
To the server, the firewall appears to be the client

Proxies

Proxy forwarding requests to and from the Internet

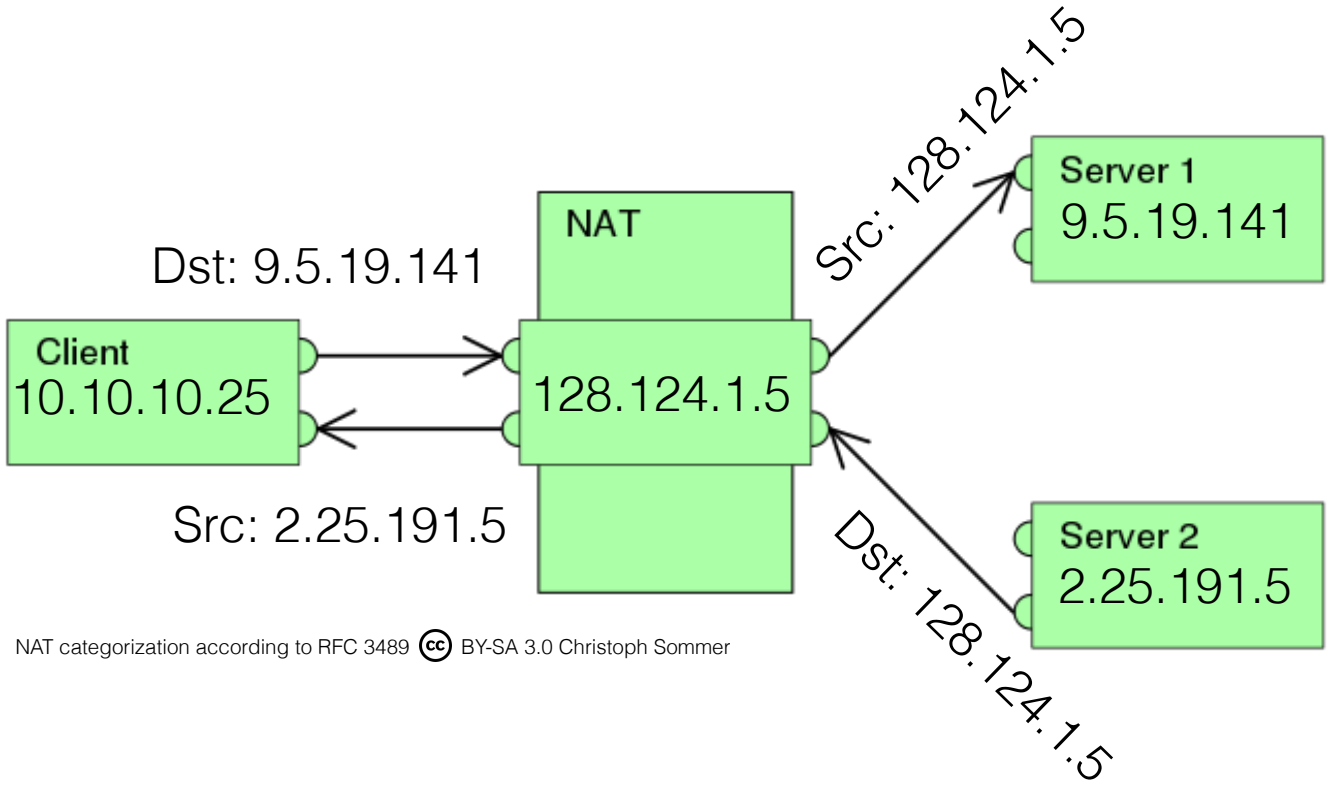


Reverse proxy takes requests from Internet and forwards them to internal network servers



Likely unaware
of internal servers

Network Address Translation



iptables

- Interface to Linux kernel firewall
  - `iptables` applies to ipv4
  - `ip6tables` applies to ipv6
  - `arptables` applies to arp
  - `ebtables` to ethernet frames

```
# iptables -L | grep policy
Chain INPUT (policy ACCEPT)
Chain FORWARD (policy ACCEPT)
Chain OUTPUT (policy ACCEPT)
```

Default Policy

https://help.ubuntu.com/community/IptablesHowTo

iptables

Three different chains for rules:

Input: controls the behavior of incoming connections

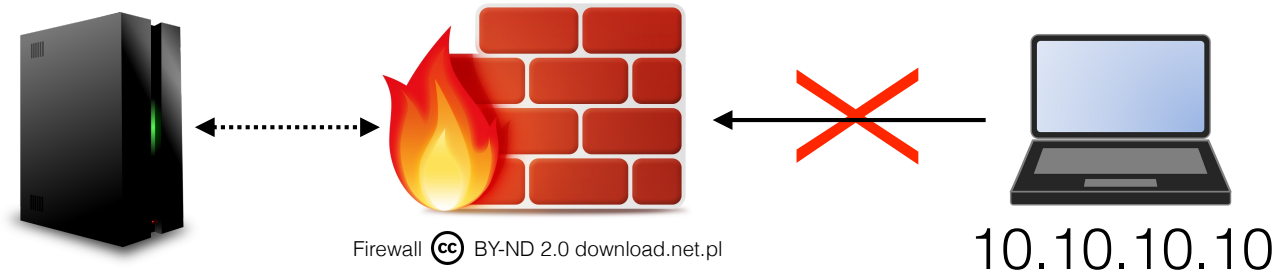
Forward: used for incoming connections that aren't being delivered locally (routing / NAT)

Output: controls the behavior of outgoing connections

`iptables`: blocking connections

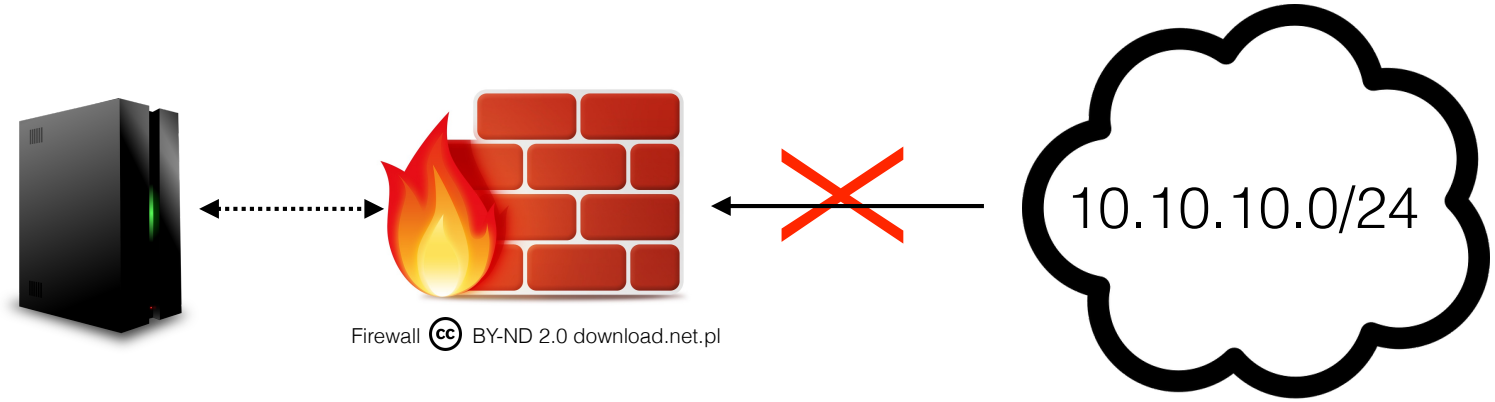
Block all connections from 10.10.10.10:

```
# iptables -A INPUT -s 10.10.10.10 -j DROP
```



Block all connections from a class C network:

```
# iptables -A INPUT -s 10.10.10.0/24 -j DROP
```



`iptables`: blocking connections

How can we stop an ssh brute force attack as it's happening?

```
sshd[22731] Failed password for invalid user admin from
10.10.10.10 port 32981
sshd[22732] Failed password for invalid user admin from
10.10.10.10 port 32989
```

...

fail2ban-style rule:

```
# iptables -A INPUT -p tcp --dport ssh -s 10.10.10.10 -j
DROP
```

`iptables`: connection states

Allow existing session to receive traffic:

```
# iptables -A INPUT -m conntrack --ctstate
ESTABLISHED,RELATED -j ACCEPT
```

Allow ssh traffic:

```
# iptables -A INPUT -p tcp --dport ssh -j ACCEPT
```

Check the current state of the rules:

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source     destination
ACCEPT     all  --  anywhere   anywhere       state
RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere   anywhere       tcp dpt:ssh
```

`iptables`: connection states

Allow www traffic:

```
# iptables -A INPUT -p tcp --dport www -j ACCEPT
```

By default, block all traffic:

```
# iptables -A INPUT -j DROP
```

Check the current state of the rules:

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source   destination
ACCEPT     all  --  anywhere anywhere state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere anywhere           tcp dpt:ssh
ACCEPT     tcp  --  anywhere anywhere           tcp dpt:www
DROP       all  --  anywhere anywhere
```

`iptables`: editing chains

We forgot to leave the loopback device open — let's fix that:

```
# iptables -I INPUT 1 -i lo -j ACCEPT
```

Check the current state of the rules:

```
# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     any     anywhere             anywhere
    0     0 ACCEPT     all  --  any    any     anywhere             anywhere
state RELATED,ESTABLISHED
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
tcp dpt:ssh
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
tcp dpt:www
    0     0 DROP       all  --  any    any     anywhere             anywhere
```

Network Intrusion Detection

IDS is a backup security mechanism

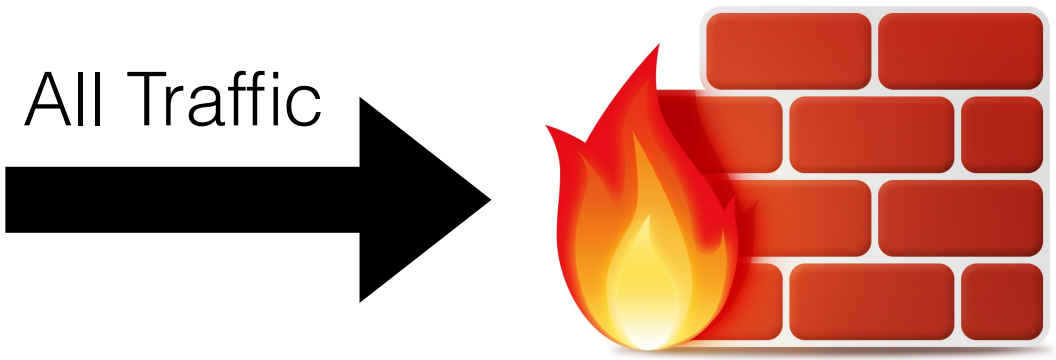
- Assumes other defenses (firewalls, hardened hosts) have failed
- Task is to notice an attack as soon as possible
    - Minimize damage
        - Automated systems or human response

Signature- or Anomaly-based

NIDS placement on network

The firewall is one common location to install a network IDS

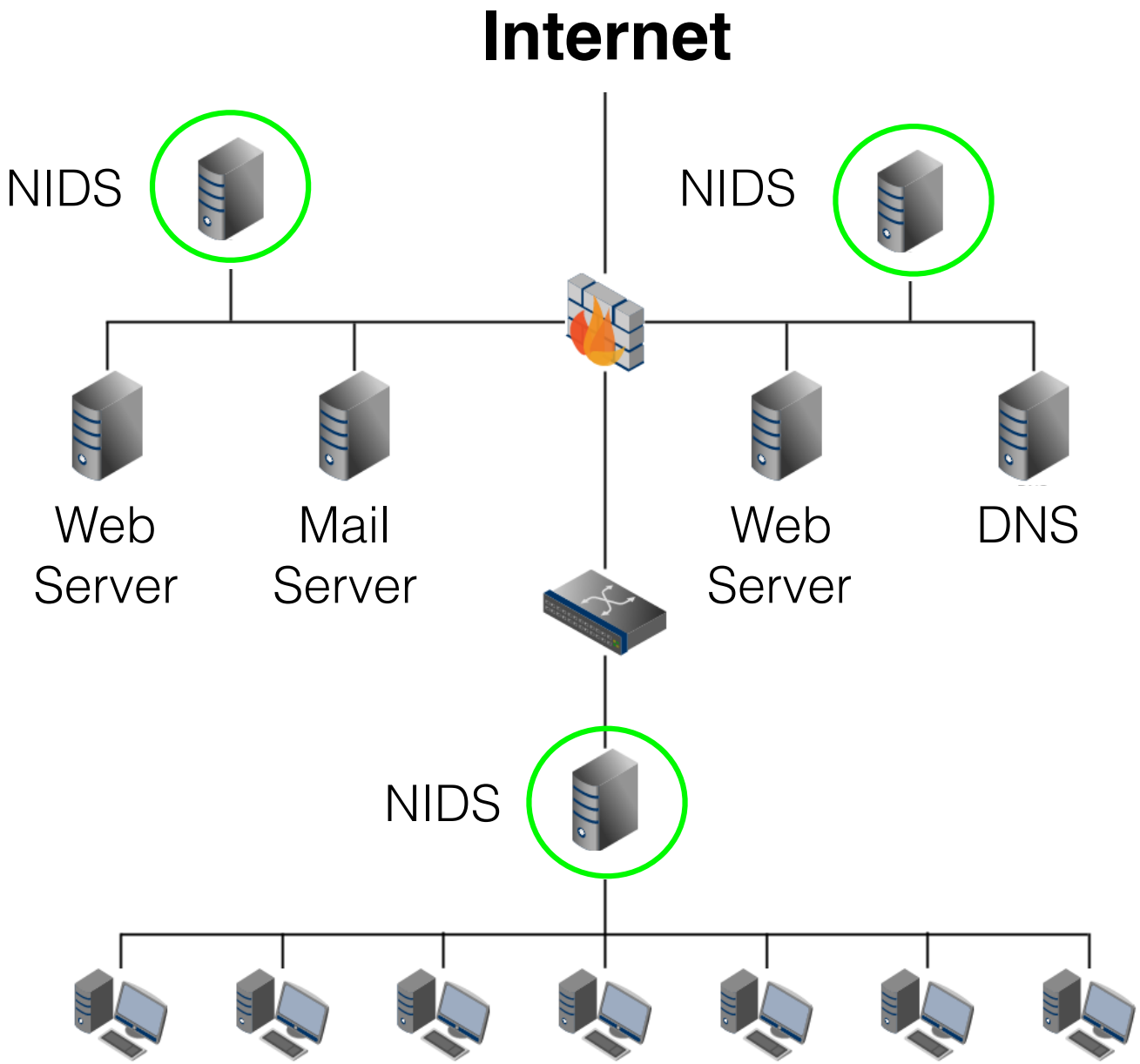
By definition, all outside traffic must pass through this chokepoint



Firewall BY-ND 2.0 download.net.pl

+ NIDS

NIDS placement on network



Signature-based IDS

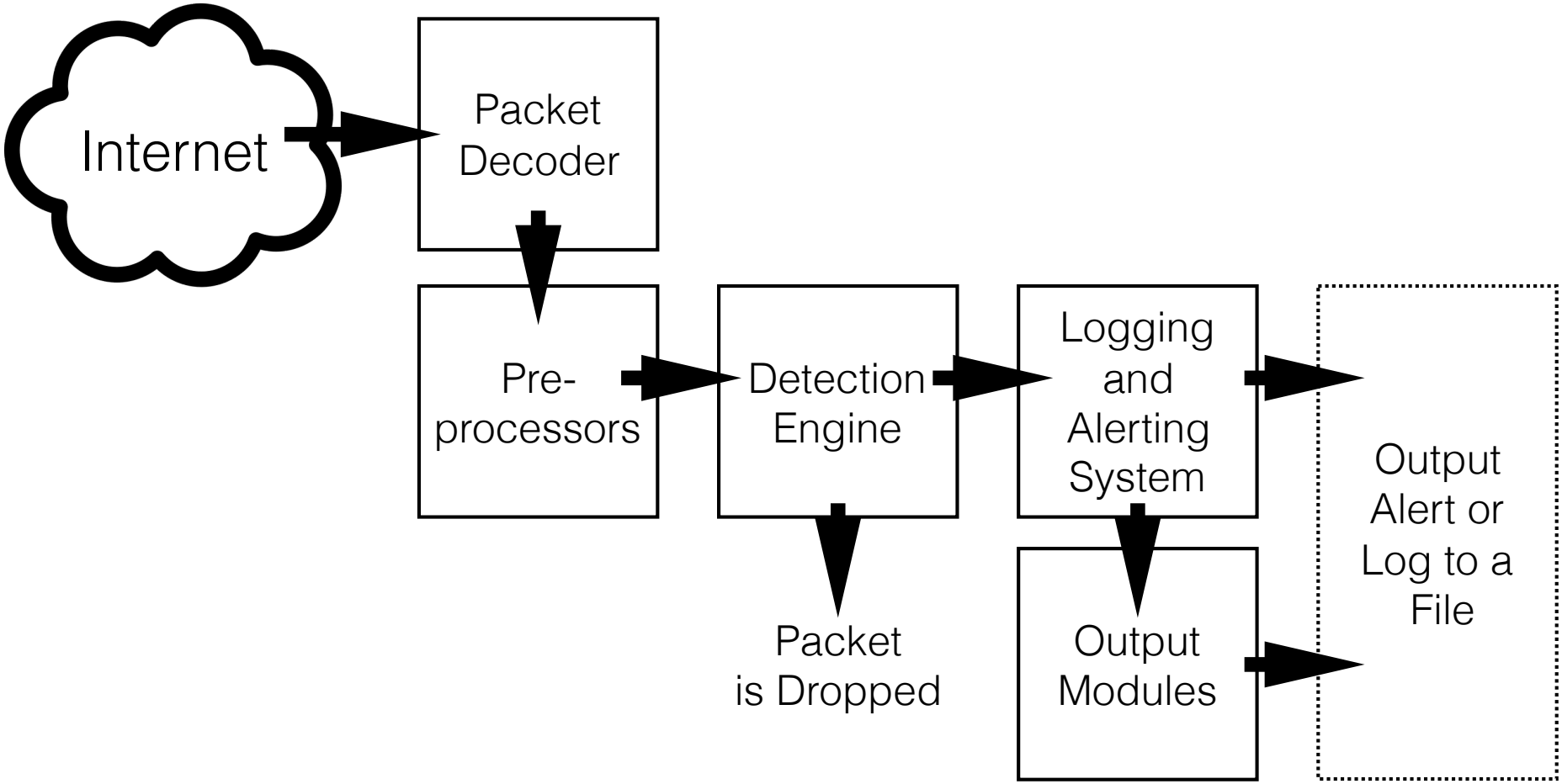
- Look for patterns in the packet headers and payloads
- Rely on pre-defined signatures for known attacks
  - Professionally developed
  - Community developed
- Examples:
  - Bro (Paxson 1998; https://www.bro.org/)
  - **Snort** (Roesch 1999)

Snort

https://www.snort.org/

Free and Open Source (commercial offerings via Sourcefire)

Snort Architecture



Usage (Unix)

Sniffer Mode

```
$ snort -vde
```
← IP/TCP/UDP/ICMP headers + data

Packet Logger Mode

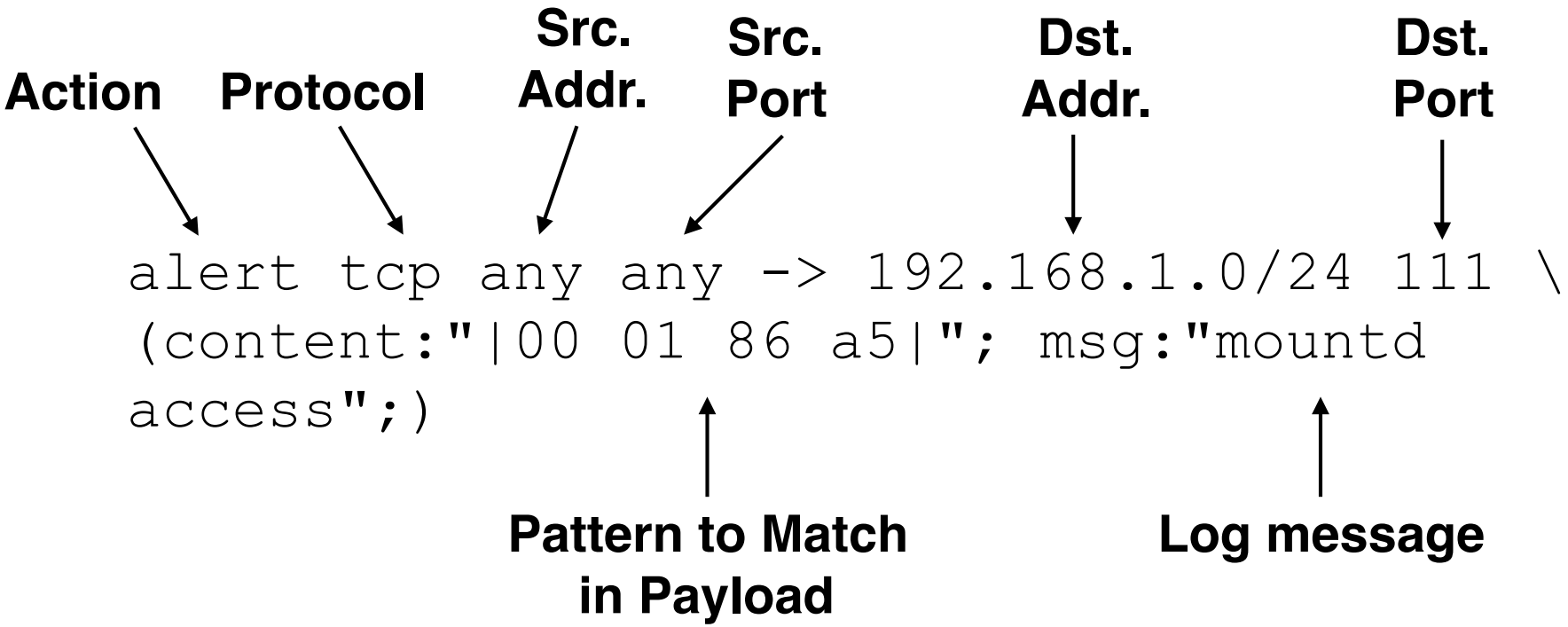
```
$ snort -dev -l ./packet.log
```
← write

```
$ snort -dv -r packet.log
```
← read

Network Intrusion Detection Mode

```
$ snort -dev -l ./log -h 192.168.1.0/24 -c
snort.conf
```
← apply signatures to traffic

Snort Signature



Snort Rules Collections

Community Rules:

https://www.snort.org/downloads/#rule-downloads

EmergingThreats.net Open rulesets:

https://rules.emergingthreats.net/open/snort-2.9.0/emerging-all.rules

Let’s examine some real-world threats and corresponding rules…

wp_advanced_custom_fields_exec

- Remote file inclusion flaw in the WordPress blogging software plugin known as Advanced Custom Fields
- The vulnerability allows for remote file inclusion and remote code execution via the `export.php` script
- The Advanced Custom Fields plug-in versions 3.5.1 and below are vulnerable
- This exploit only works when the php option `allow_url_include` is set to On (Default Off).

(exploit available in kali linux 2.0)

wp_advanced_custom_fields_exec

```
12:39:49.945748 IP 192.168.56.102.41634 > 192.168.56.101.80: Flags [P.],
seq 1:293, ack 1, win 29, options [nop,nop,TS val 485350 ecr 487168],
length 292
    0x0000:  4500 0158 5168 4000 4006 f61b c0a8 3866     E..XQh@.@.....8f
    0x0010:  c0a8 3865 a2a2 0050 4db3 3681 d50f 9d27     ..8e...PM.6....'
    0x0020:  8018 001d 4657 0000 0101 080a 0007 67e6     ....FW........g.
    0x0030:  0007 6f00 504f 5354 202f 7770 2d63 6f6e     ..o.POST./wp-con
    0x0040:  7465 6e74 2f70 6c75 6769 6e73 2f61 6476     tent/plugins/adv
    0x0050:  616e 6365 642d 6375 7374 6f6d 2d66 6965     anced-custom-fie
    0x0060:  6c64 732f 636f 7265 2f61 6374 696f 6e73     lds/core/actions
    0x0070:  2f65 7870 6f72 742e 7068 7020 4854 5450     /export.php.HTTP
    0x0080:  2f31 2e31 0d0a 486f 7374 3a20 3139 322e     /1.1..Host:.192.
    0x0090:  3136 382e 3536 2e31 3031 0d0a 5573 6572     168.56.101..User
    0x00a0:  2d41 6765 6e74 3a20 4d6f 7a69 6c6c 612f     -Agent:.Mozilla/
    0x00b0:  342e 3020 2863 6f6d 7061 7469 626c 653b     4.0.(compatible;
    0x00c0:  204d 5349 4520 362e 303b 2057 696e 646f     .MSIE.6.0;.Windo
    0x00d0:  7773 204e 5420 352e 3129 0d0a 436f 6e74     ws.NT.5.1)..Cont
    0x00e0:  656e 742d 5479 7065 3a20 6170 706c 6963     ent-Type:.applic
    0x00f0:  6174 696f 6e2f 782d 7777 772d 666f 726d     ation/x-www-form
    0x0100:  2d75 726c 656e 636f 6465 640d 0a43 6f6e     -urlencoded..Con
    0x0110:  7465 6e74 2d4c 656e 6774 683a 2035 330d     tent-Length:.53.
    0x0120:  0a0d 0a61 6366 5f61 6273 7061 7468 3d68     ...acf_abspath=h
    0x0130:  7474 703a 2f2f 3139 322e 3136 382e 3536     ttp://192.168.56
    0x0140:  2e31 3031 3a38 3038 302f 574d 3938 7934     .101:8080/WM98y4
    0x0150:  6852 7263 3646 653f                         hRrc6Fe?
```

wp_advanced_custom_fields_exec

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"ET WEB_SPECIFIC_APPS WordPress Plugin Advanced Custom
Fields Remote File Inclusion"; flow:established,to_server;
content:"/wp-content/plugins/advanced-custom-fields/core/
actions/export.php"; nocase; http_uri; fast_pattern:20,20;
content:"abspath="; nocase; http_client_body; pcre:"/
abspath=\s*(?:(?:ht|f)tps?|data|php)\x3a\//Pi";
classtype:attempted-user; sid:2016148; rev:1;)
```

Wuerzburg Shellcode

Origin: Nepenthes modular low-interaction honeypot

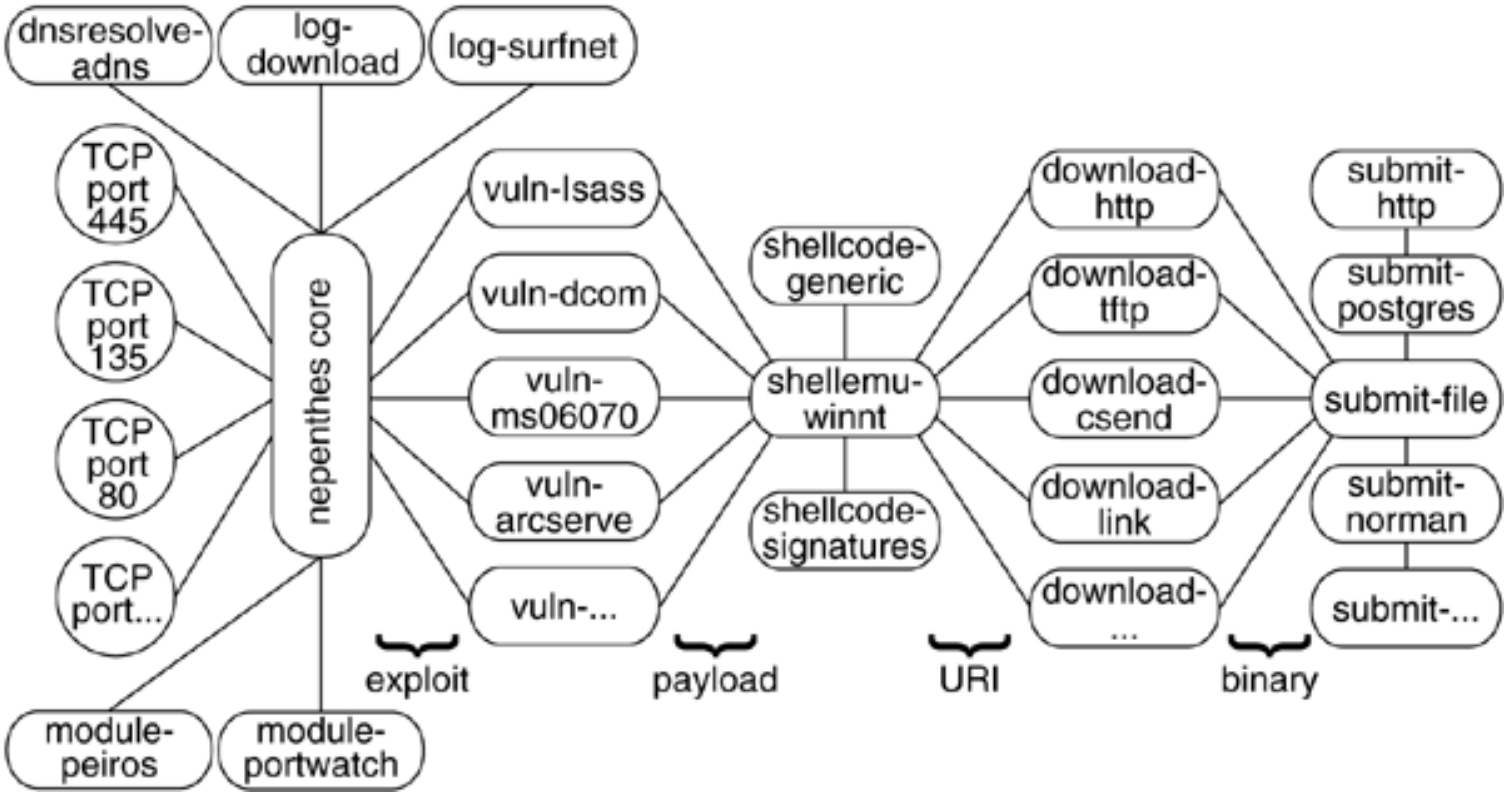


Image source: http://books.gigatux.nl/mirror/honeypot/final/ch06lev1sec2.html

Wuerzburg Shellcode

```
0040200c   eb 27            jmp short wuerzbur.00402035
0040200e   90               nop
0040200f   90               nop
00402010   90               nop
00402011   90               nop
00402012   90               nop
00402013   90               nop
00402014   5d               pop ebp
00402015   33c9             xor ecx,ecx
00402017   66:b9 2502       mov cx,225
0040201b   8d75 05          lea esi,dword ptr ss:[ebp+5]
0040201e   8bfe             mov edi,esi
00402020   8a06             mov al,byte ptr ds:[esi]
00402022   3c 99            cmp al,99
00402024   75 05            jnz short wuerzbur.0040202b
00402026   46               inc esi
00402027   8a06             mov al,byte ptr ds:[esi]
00402029   2c 30            sub al,30
0040202b   46               inc esi
0040202c   34 99            xor al,99
0040202e   8807             mov byte ptr ds:[edi],al
00402030   47               inc edi
00402031  ^e2 ed            loopd short wuerzbur.00402020
00402033   eb 0a            jmp short wuerzbur.0040203f
00402035   e8 daffffff      call wuerzbur.00402014
```

Wuerzburg Shellcode

```
alert tcp any any -> any any (msg:"ET SHELLCODE
Wuerzburg Shellcode"; flow:established; content:"|
eb 27|"; content:"|5d 33 c9 66 b9|"; distance:0;
content:"|8d 75 05 8b fe 8a 06 3c|"; distance:0;
content:"|75 05 46 8a 06|"; distance:0; content:"|
88 07 47 e2 ed eb 0a e8 da ff ff ff|"; distance:0;
reference:url,doc.emergingthreats.net/2009251;
classtype:shellcode-detect; sid:2009251; rev:3;)
```

NOP detection (ssh exploit)

```
#alert tcp $EXTERNAL_NET any -> $HOME_NET 22
(msg:"GPL SHELLCODE ssh CRC32 overflow
NOOP"; flow:to_server,established;
content:"|90 90 90 90 90 90 90 90 90 90
90 90 90 90 90|"; reference:bugtraq,2347;
reference:cve,2001-0144; reference:cve,
2001-0572; classtype:shellcode-detect; sid:
2101326; rev:7;)
```

Metasploit Meterpreter Kill Process

Meterpreter: advanced, dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime

Provides a basic shell and allows new features to be added as necessary

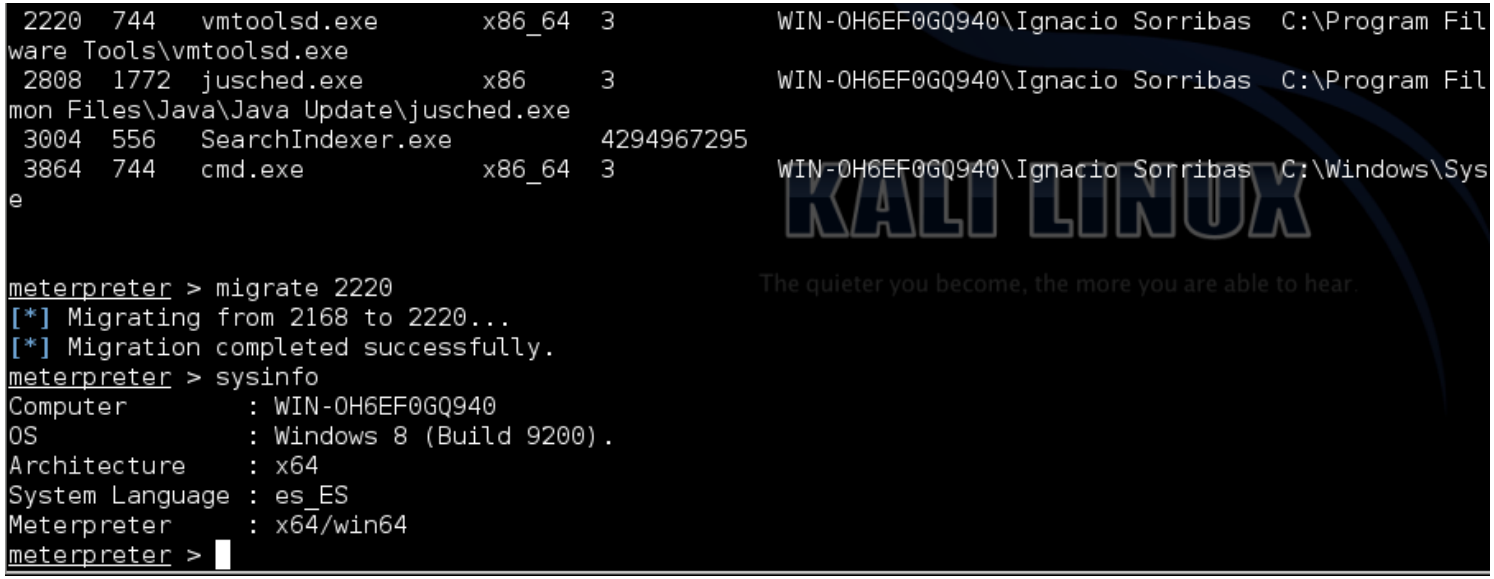


Image credit: http://hardsec.net/wp-content/uploads/2014/01/sniffer2.png

Metasploit Meterpreter Kill Process

Ruby code:

```
def Process.kill(*args)
    request =
      Packet.create_request('stdapi_sys_process_kill')

    args.each { |id|
       request.add_tlv(TLV_TYPE_PID, id)
    }

    client.send_request(request)

    return true
end
```

Metasploit Meterpreter Kill Process

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any
(msg:"ET ATTACK_RESPONSE Metasploit Meterpreter Kill
Process Command Detected";
flow:to_client,established;
content:"stdapi_sys_process_kill"; depth:60;
reference:url,www.nologin.org/Downloads/Papers/
meterpreter.pdf;
reference:url,doc.emergingthreats.net/2009565;
classtype:successful-user; sid:2009565; rev:4;)
```

nmap XMAS scan

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any
(msg:"GPL SCAN nmap XMAS"; flow:stateless;
flags:FPU,12; reference:arachnids,30;
classtype:attempted-recon; sid:2101228; rev:8;)
```

nmap SYN scan

```
alert tcp $EXTERNAL_NET any -> $HOME_NET
any (msg:"ET SCAN NMAP -sS window 1024";
fragbits:!D; dsize:0; flags:S,12; ack:0;
window:1024; threshold: type both, track
by_dst, count 1, seconds 60;
reference:url,doc.emergingthreats.net/
2009582; classtype:attempted-recon; sid:
2009582; rev:3;)
```

Rapid IMAP Connections

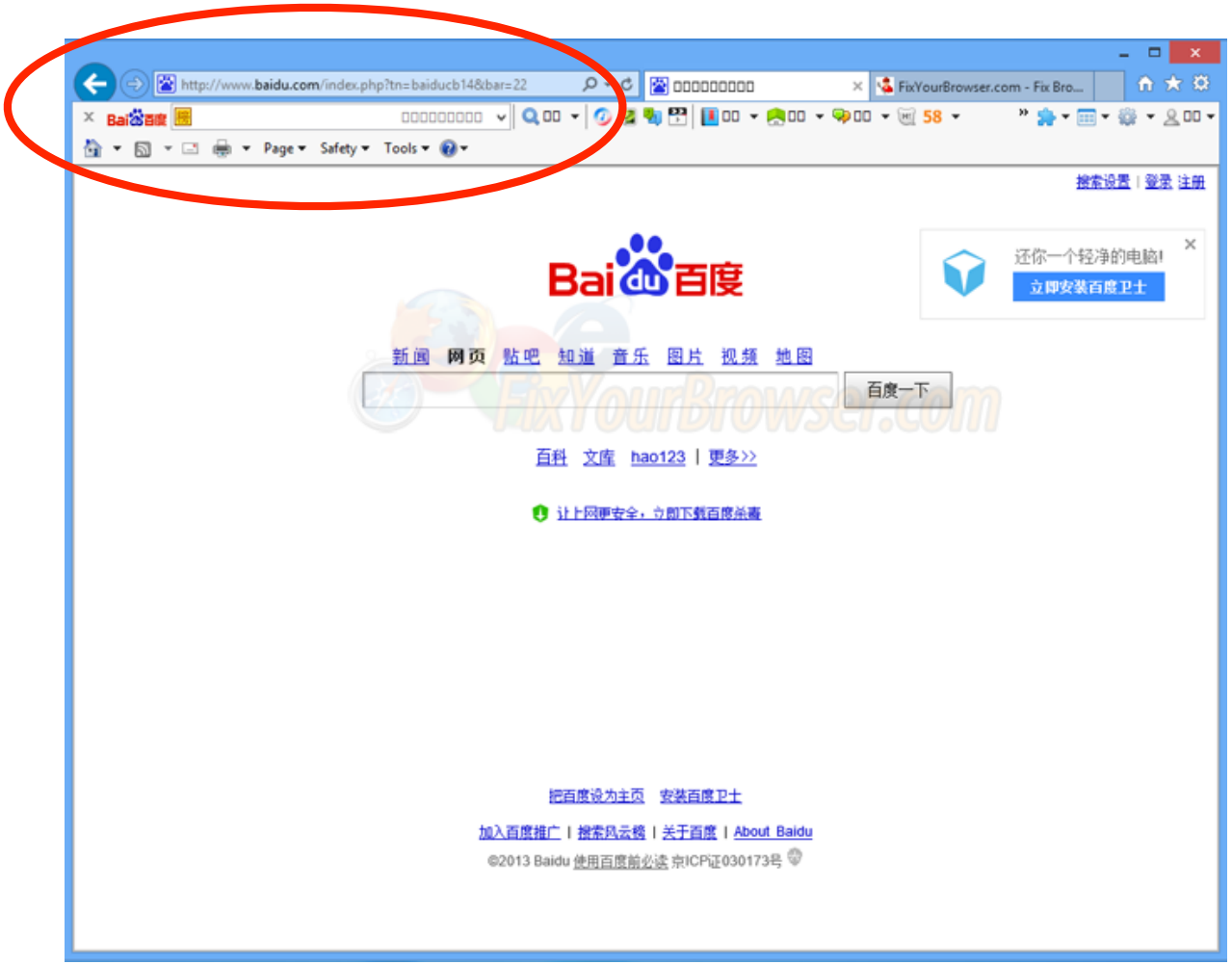
```
alert tcp $EXTERNAL_NET any -> $HOME_NET 143
(msg:"ET SCAN Rapid IMAP Connections - Possible
Brute Force Attack"; flags: S,12; threshold:
type both, track by_src, count 30, seconds 60;
reference:url,doc.emergingthreats.net/2002994;
classtype:misc-activity; sid:2002994; rev:6;)
```

Baidu Toolbar

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"ET MALWARE Baidu.com Spyware Bar Pulling Data";
flow:to_server,established; content:"/cpro/ui/ui"; nocase;
http_uri; content:"baidu.com"; nocase; http_header;
content:!"Referer|3a| "; nocase; http_header;
reference:url,www.pctools.com/mrc/infections/id/BaiDu/;
reference:url,doc.emergingthreats.net/bin/view/Main/
2003578; classtype:trojan-activity; sid:2003578; rev:9;)
```

Baidu Toolbar

Origin of this toolbar is somewhat unclear



https://www.fixyourbrowser.com/removal-instructions/remove-baidu-toolbar/

Fake updates for Windows

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any
(msg:"ET CURRENT_EVENTS Fake MS Security Update
(Jar)"; flow:established,from_server; file_data;
content:"Microsoft Security Update";
content:"applet_ssv_validated"; fast_pattern:only;
flowbits:set,et.exploitkitlanding; classtype:trojan-
activity; sid:2017549; rev:1;)
```

SQL Injection

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS
$HTTP_PORTS (msg:"ET WEB_SPECIFIC_APPS Kubix SQL
Injection Attempt -- index.php member_id SELECT";
flow:established,to_server; content:"/index.php?";
nocase; http_uri; content:"member_id="; nocase;
http_uri; content:"SELECT"; nocase; http_uri; pcre:"/
SELECT.+FROM/Ui"; reference:cve,CVE-2006-7116;
reference:url,www.exploit-db.com/exploits/2863/;
reference:url,doc.emergingthreats.net/2004689;
classtype:web-application-attack; sid:2004689; rev:
7;)
```

SQL Injection in Cookie

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"ET WEB_SERVER Possible DELETE FROM SQL Injection In
Cookie"; flow:to_server,established; content:"DELETE ";
nocase; http_cookie; content:"FROM"; nocase; http_cookie;
pcre:"/\x0a\x0dCookie\x3a[^\n]DELETE.+FROM/i";
reference:url,www.w3schools.com/Sql/sql_delete.asp;
reference:url,en.wikipedia.org/wiki/SQL_injection;
reference:url,www.owasp.org/index.php/SQL_Injection;
reference:url,doc.emergingthreats.net/2009772;
classtype:web-application-attack; sid:2009772; rev:7;)
```

Heartbleed

```
alert tcp any any -> $HOME_NET !$HTTP_PORTS (msg:"ET
CURRENT_EVENTS Malformed HeartBeat Request";
flow:established,to_server; content:"|18 03|";
depth:2; byte_test:1,<,4,2; content:"|01|"; offset:
5; depth:1; byte_extract:2,3,record_len; byte_test:
2,>,2,3; byte_test:2,>,record_len,6; threshold:type
limit,track by_src,count 1,seconds 120;
flowbits:set,ET.MalformedTLSHB; reference:cve,
2014-0160; reference:url,blog.inliniac.net/
2014/04/08/detecting-openssl-heartbleed-with-
suricata/; reference:url,heartbleed.com/;
reference:url,blog.fox-it.com/2014/04/08/openssl-
heartbleed-bug-live-blog/; classtype:bad-unknown;
sid:2018372; rev:2;)
```

Heartbleed 

```
alert tcp $HOME_NET !$HTTP_PORTS -> any any (msg:"ET
CURRENT_EVENTS Malformed HeartBeat Response";
flow:established,from_server;
flowbits:isset,ET.MalformedTLSHB; content:"|18 03|";
depth:2; byte_test:1,<,4,2; byte_test:2,>,200,3;
threshold:type limit,track by_src,count 1,seconds 120;
reference:cve,2014-0160;
reference:url,blog.inliniac.net/2014/04/08/detecting-
openssl-heartbleed-with-suricata/;
reference:url,heartbleed.com/; reference:url,blog.fox-
it.com/2014/04/08/openssl-heartbleed-bug-live-blog/;
classtype:bad-unknown; sid:2018373; rev:3;)
```

Tor Detection

```
alert tcp $HOME_NET any -> $EXTERNAL_NET
$HTTP_PORTS (msg:"ET POLICY Onion2Web Tor Proxy
Cookie"; flow:established,to_server;
content:"onion2web_confirmed="; fast_pattern:only;
content:"onion2web_confirmed="; http_cookie;
reference:md5,a46e609662eb94a726fcb4471b7057d4;
reference:md5,2b62cdb6bcec4bff47eff437e4fc46d3;
reference:url,github.com/starius/onion2web;
classtype:policy-violation; sid:2020324; rev:1;)
```

(Not Very) Covert Channel

```
alert udp $HOME_NET any -> any 53 (msg:"ET TROJAN
Large DNS Query possible covert channel"; content:"|01
00 00 01 00 00 00 00 00 00|"; fast_pattern; depth:10;
offset:2; dsize:>300; content:!"youtube|03|com|00|";
content:!"sophosxl|03|net|00|"; content:!"|0a|
hashserver|02|cs|0a|trendmicro|03|com|00|";
content:!"spamhaus|03|org|00|"; classtype:bad-unknown;
sid:2013075; rev:8;)
```

Denial of Service Detection

```
alert tcp $EXTERNAL_NET 10000: -> $HOME_NET
0:1023 (msg:"ET DOS Potential Tsunami SYN Flood
Denial Of Service Attempt"; flags:S; dsize:>900;
threshold: type both, count 20, seconds 120,
track by_src; reference:url,security.radware.com/
uploadedFiles/Resources_and_Content/Threat/
TsunamiSYNFloodAttack.pdf; classtype:attempted-
dos; sid:2019404; rev:2;)
```

High false positive potential

Out-of-date Software Detection

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"ET POLICY Vulnerable Java Version 1.8.x
Detected"; flow:established,to_server; content:"
Java/1.8.0_"; http_header; content:!"73"; within:2;
http_header; content:!"74"; within:2; http_header;
flowbits:set,ET.http.javaclient.vulnerable;
threshold: type limit, count 2, seconds 300, track
by_src; reference:url,javatester.org/version.html;
classtype:bad-unknown; sid:2019401; rev:9;)
```