



Welcome to

Intrusion Detection Basics

Making sense of packets

Henrik Kramselund Jereminsen hkj@zecurity.com @kramse  

Slides are available as PDF, [kramse@Github
intrusion-detection-basics.tex](https://github.com/kramse/security-courses/blob/main/intrusion-detection-basics.tex) in the repo security-courses

Contact information



- Henrik Kramselund Jereminsen, internet samurai mostly networks and infosec
- Network and security consultant Zecurity, teach at KEA and activist
- Master from the Computer Science Department at the University of Copenhagen, DIKU
- Email: hkj@zecurity.dk Mobile: +45 2026 6000

You are welcome to drop me an email

Goals for today



The Zeek Network Security Monitor

See how to sniff a few packets from popular protocols

See how sniffing can be automated using two example tools Zeek and Suricata

Intrusion Detection Systems

Use Ansible to provision services - which can easily be modified to cover multiple servers

See a Kibana dashboard or two

Exercises – which you can do at home later

- Run Zeek and Suricata on small pcaps

Time schedule



- 17:00 - 18:15 Introduction and basic packet tools
- 30min break
- 18:45 - 19:30 45min Recommended tools, Zeek and Suricata – together
- 15min break
- 19:45 - 21:00 Centralized solutions dashboards with break somewhere

Duration: 4 hours - with breaks

Keywords: TCP, UDP, ICMP, TLS, DNS, Sniffing networks, malware detection, IDS, Netflow, dashboards, , Tap/SPAN/Mirror ports, Zeek Security Monitor, Suricata with SELKS, Kibana, Emerging Threats, VXLAN, Kubernetes

About equipment and exercises



- Bringing a laptop to my courses is not required, but welcome
- Links etc. are in the slides and open source licensed, PDFs
- Exercises booklets are available for many of my courses, see Github
but it is expected that participants will do any exercises on their own later or at the scheduled hacker days
- The hacker days will be announced in various places
- Events like BornHack are excellent places to arrange hacker days in the network warrior village, or other places

Invite a few friends, make a hacker day and work together!

Course Materials



This material is in multiple parts:

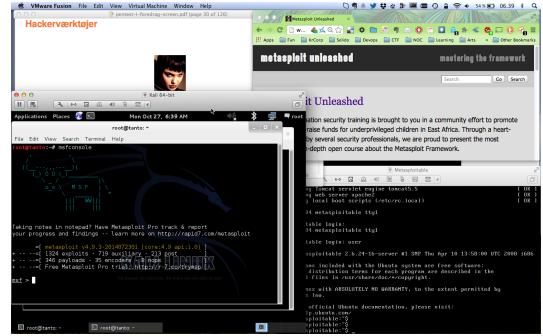
- Slide shows - presentation - this file
- Exercises - PDF files in my repository

Links

- All materials will be released as open source at:
<https://github.com/kramse/security-courses/>
- Additional resources from the internet linked from lecture plans:
<https://zencurity.gitbook.io/kea-it-sikkerhed/>

Note: slides and materials will mostly be in english, but presentation language will be danish

Hackerlab Setup



- Hardware: modern laptop CPU with virtualisation
Dont forget to enable hardware virtualisation in the BIOS
- Software Host OS: Windows, Mac, Linux
- Virtualisation software: VMware, Virtual box, HyperV pick your poison
- Hackersoftware: Kali Virtual Machine <https://www.kali.org/>
- Production use: Debian Linux <https://www.debian.org/>

Networking Hardware



If you want to do exercises and work with networks

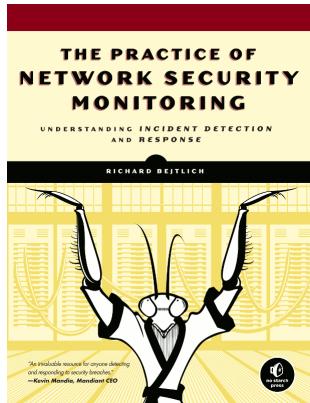
It will be an advantage to have a wireless USB network card and an USB Ethernet card.

The following are two models I have used:

- TP-link TL-WN722N hardware version 2.0 cheap
- Alfa AWUS036ACH 2.4GHz + 5GHz Dual-Band and high performing
- Often you need to compile drivers yourself, and research a bit

Getting an USB card allows you to use the regular one for the main OS, and insert the USB into the virtual machine

Network Security Monitoring – Intrusion Detection



Network Security Monitoring (NSM) - monitoring networks for intrusions, and reacting to those networkbased intrusion detection systems (NIDS)
host based intrusion detection systems (HIDS)

Example full systems are Security Onion <https://securityonion.net/> or
SELKS <https://www.stamus-networks.com/open-source/>

False Positives



- True Positive (TP). An alert that has correctly identified a specific activity. If a signature was designed to detect a certain type of malware, and an alert is generated when that malware is launched on a system, this would be a true positive, which is what we strive for with every deployed signature. *Indicators of Compromise and Signatures*
- False Positive (FP). An alert has incorrectly identified a specific activity. If a signature was designed to detect a specific type of malware, and an alert is generated for an instance in which that malware was not present, this would be a false positive.
- True Negative (TN). An alert has correctly not been generated when a specific activity has not occurred. If a signature was designed to detect a certain type of malware, and no alert is generated without that malware being launched, then this is a true negative, which is also desirable. This is difficult, if not impossible, to quantify in terms of NSM detection.
- False Negative (FN). An alert has incorrectly not been generated when a specific activity has occurred.

Source: Applied Network Security Monitoring Collection, Detection, and Analysis, 2014 Chris Sanders

Detection Capabilities



Security incidents happen, but what happens. One of the actions to reduce impact of incidents are done in preparing for incidents.

Preparation for an attack, establish procedures and mechanisms for detecting and responding to attacks

Preparation will enable easy **identification** of affected systems, better **containment** which systems are likely to be infected, **eradication** what happened – how to do the **eradication** and **recovery**.

Indicators of Compromise and Signatures



An IOC is any piece of information that can be used to objectively describe a network intrusion, expressed in a platform-independent manner. This could include a simple indicator such as the IP address of a command and control (C2) server or a complex set of behaviors that indicate that a mail server is being used as a malicious SMTP relay.

When an IOC is taken and used in a platform-specific language or format, such as a Snort Rule or a Bro-formatted file, it becomes part of a signature. A signature can contain one or more IOCs.

Source: Applied Network Security Monitoring Collection, Detection, and Analysis, 2014 Chris Sanders

What happens today?



Think like a blue team member find hacker traffic

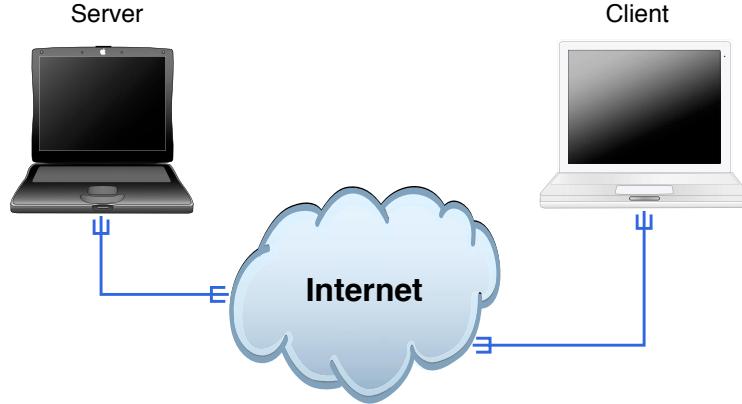
Get basic tools running

Improve situation

- See where the data end up
- What kind of data and metadata can we extract
- How can we collect and make use of it
- Databases and web interfaces, examples shown
- Consider what your deployment could be

Today focus on the lower parts, but user interfaces are important too

Internet Today



- Clients and servers, roots in the academic world
- Protocols are old, some more than 20 years
- Very little is encrypted, mostly HTTPS

What is the Internet



Communication between humans - currently!

Based on TCP/IP

- best effort
- packet switching (IPv6 calls it packets, not datagram)
- *connection-oriented* TCP
- *connection-less* UDP

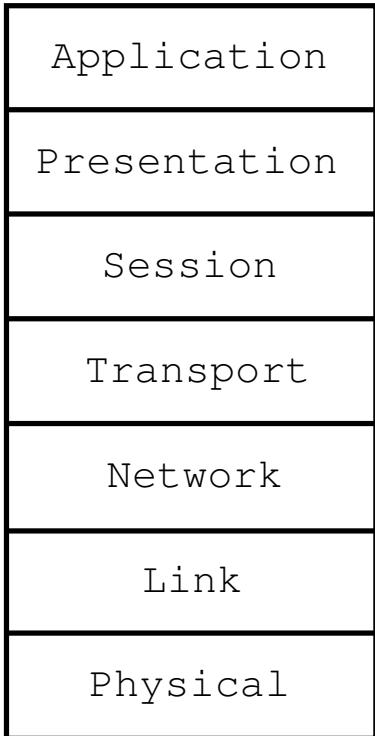
RFC-1958:

A good analogy for the development of the Internet is that of constantly renewing the individual streets and buildings of a city, rather than razing the city and rebuilding it. The architectural principles therefore aim to provide a framework for creating cooperation and standards, as a small "spanning set" of rules that generates a large, varied and evolving space of technology.

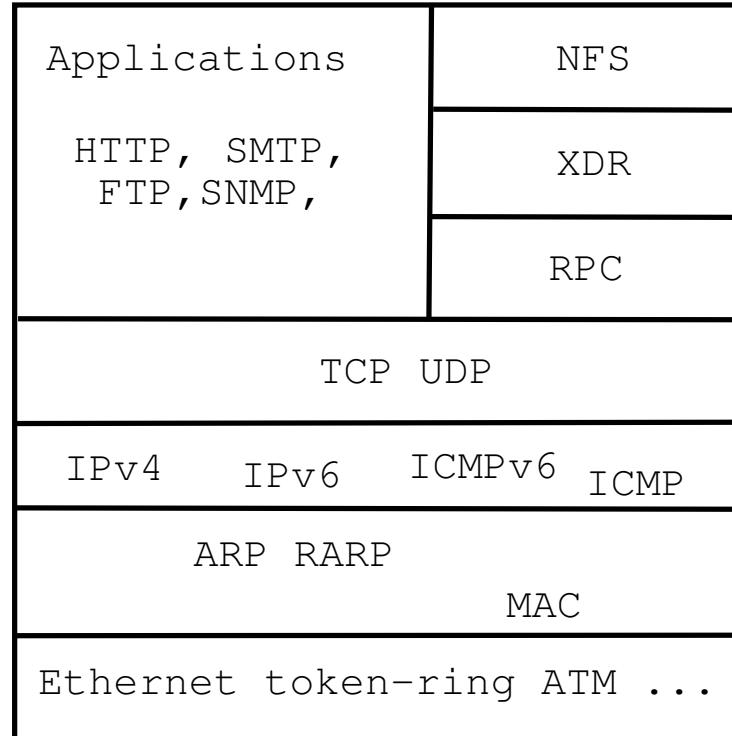
OSI and Internet models



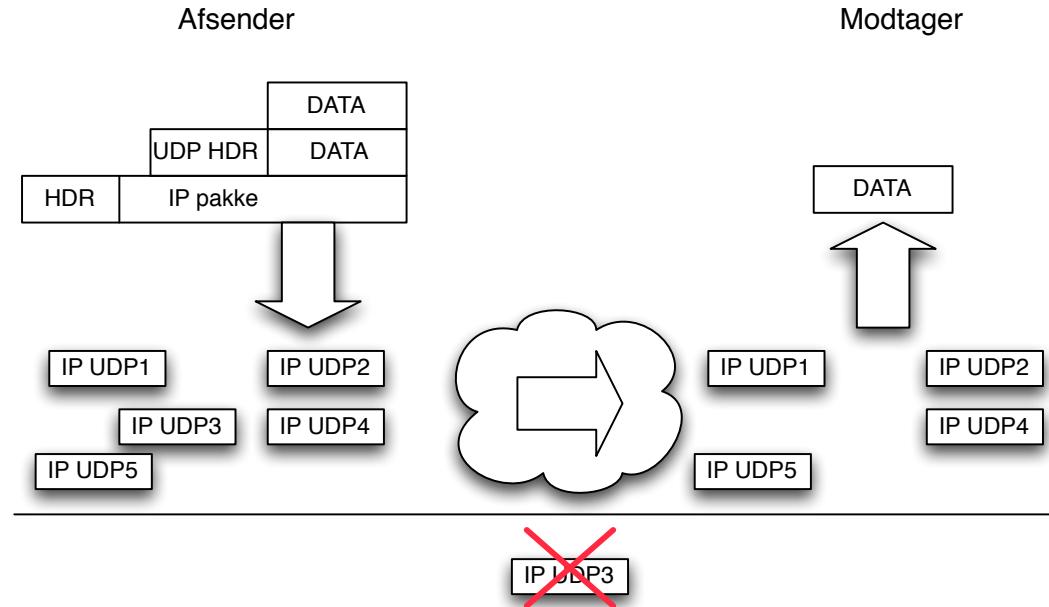
OSI Reference Model



Internet protocol suite



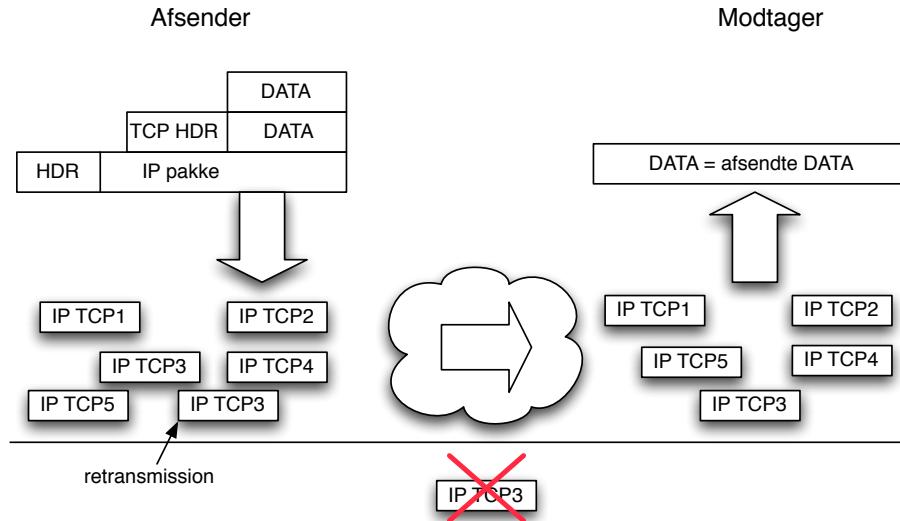
UDP User Datagram Protocol



Connection-less RFC-768, *connection-less*

Used for Domain Name Service (DNS)

TCP Transmission Control Protocol



Connection oriented RFC-791 September 1981, *connection-oriented*

Either data delivered in correct order, no data missing, checksum or an error is reported

Used for HTTP and others

Well-known port numbers



IANA maintains a list of magical numbers in TCP/IP
Lists of protocol numbers, port numbers etc.

A few notable examples:

- Port 25/tcp Simple Mail Transfer Protocol (SMTP)
- Port 53/udp and 53/tcp Domain Name System (DNS)
- Port 80/tcp Hyper Text Transfer Protocol (HTTP)
- Port 443/tcp HTTP over TLS/SSL (HTTPS)

Source: <http://www.iana.org>

ICMP Internet Control Message Protocol



Control protocol, error messages

Common messages

- ICMP ECHO, anyone there?
- Host unreachable
- Port unreachable

signaling

Defined in RFC-792

IPv6 neighbor discovery protocol (NDP)



| OSI | IPv4 | IPv6 |
|----------|-----------|---------------|
| Network | IP / ICMP | IPv6 / ICMPv6 |
| Link | ARP | |
| Physical | Physical | Physical |

Address Resolution Protocol (ARP) is gone from IPv6

NDP replaces and expands, command to use arp -an replaced by ndp -an

Basic test tools TCP - Ping and Traceroute



We should all know

- ping – like sending a radar ping, anything there
- traceroute (windows tracert) – find the route packets traverse

and add these!

- Wireshark – like sending a radar ping, anything there
- Nmap and Nping – port scan and advanced ping program!

Ping



```
$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=150 time=8.849 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=150 time=0.588 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=150 time=0.553 ms
```

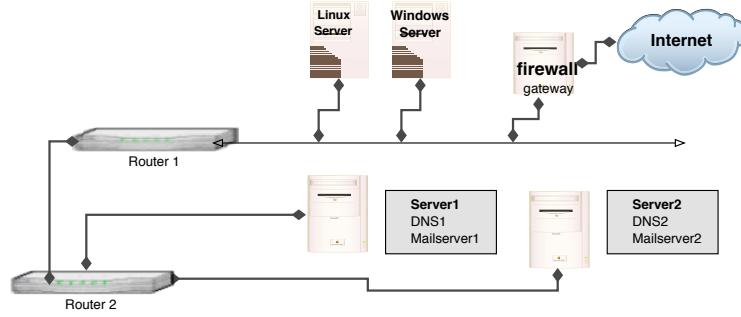
ICMP – Internet Control Message Protocol

ECHO – only ICMP message that generates another

ICMP ECHO request sent, and ICMP ECHO reply expected

Same with IPv6, ping6

Network mapping



Using traceroute and similar programs it is often possible to make educated guess to network topology

Time to live (TTL) for packets are decreased when crossing a router
when it reaches zero the packet is timed out, and ICMP message sent back to source
Default Unix traceroute uses UDP, Windows tracert use ICMP

traceroute



```
hkj@bob:~$ traceroute www.kramse.dk
traceroute to www.kramse.dk (185.129.60.130), 30 hops max, 60 byte packets
 1  10.0.42.1 (10.0.42.1)  0.365 ms  0.277 ms  0.239 ms
 2  79.142.xxx.xxx (79.142.xxx.xxx)  5.174 ms  4.979 ms  5.113 ms
 3  bgp2-dix.prod.bolignet.dk (79.142.224.2)  5.538 ms  5.057 ms  5.483 ms
 4  217.74.215.57 (217.74.215.57)  5.990 ms  5.962 ms  5.932 ms
...
 8  185.150.199.178 (185.150.199.178)  7.684 ms  7.647 ms  4.627 ms
 9  * * * // firewall here!
```

Works using the Time to live (TTL) counter

Sending with $TTL = 1$ returns ICMP from first host/router

Default sends UDP on Unix, and ICMP on Windows

Kali has programs that can emulate, or send using any protocol

traceroute – UDP



```
# tcpdump -i en0 host 10.20.20.129 or host 10.0.0.11
tcpdump: listening on en0
23:23:30.426342 10.0.0.200.33849 > router.33435: udp 12 [ttl 1]
23:23:30.426742 safri > 10.0.0.200: icmp: time exceeded in-transit
23:23:30.436069 10.0.0.200.33849 > router.33436: udp 12 [ttl 1]
23:23:30.436357 safri > 10.0.0.200: icmp: time exceeded in-transit
23:23:30.437117 10.0.0.200.33849 > router.33437: udp 12 [ttl 1]
23:23:30.437383 safri > 10.0.0.200: icmp: time exceeded in-transit
23:23:30.437574 10.0.0.200.33849 > router.33438: udp 12
23:23:30.438946 router > 10.0.0.200: icmp: router udp port 33438 unreachable
23:23:30.451319 10.0.0.200.33849 > router.33439: udp 12
23:23:30.452569 router > 10.0.0.200: icmp: router udp port 33439 unreachable
23:23:30.452813 10.0.0.200.33849 > router.33440: udp 12
23:23:30.454023 router > 10.0.0.200: icmp: router udp port 33440 unreachable
23:23:31.379102 10.0.0.200.49214 > safri.domain: 6646+ PTR?
200.0.0.10.in-addr.arpa. (41)
23:23:31.380410 safri.domain > 10.0.0.200.49214: 6646 NXDomain* 0/1/0 (93)
14 packets received by filter
0 packets dropped by kernel
```

Low TTL, UDP, high ports above 33000 = Unix traceroute signature

Packet sniffing tools



Tcpdump for capturing packets

Wireshark for dissecting packets manually with GUI

Wireshark - graphical network sniffer



We're having a conference! You're invited!

Download
Get Started Now

Learn
Knowledge is Power

Enhance
With Riverbed Technology

News And Events

Join us at SHARKFEST '15!

SHARKFEST '15 will be held from June 22 – 25 at the Computer History Museum in Mountain View, CA.

[Learn More ▶](#)

Troubleshooting with Wireshark

By Laura Chappell
Foreword by Gerald Combs
Edited by Jim Aragon

This book focuses on the tips and techniques used to identify

Wireshark Blog

Cool New Stuff

Dec 17 | By Evan Huus

Wireshark 1.12 Officially Released!

Jul 31 | By Evan Huus

To Infinity and Beyond! Capturing Forever with Tshark

Jul 8 | By Evan Huus

[More Blog Entries ▶](#)

Enhance Wireshark

Riverbed is Wireshark's primary sponsor and provides our funding. They also make great products.

802.11 Packet Capture

- WLAN packet capture and transmission
- Full 802.11 a/b/g/n support
- View management, control and data frames
- Multi-channel aggregation (with multiple adapters)

[Learn More ▶](#)

[Buy Now ▶](#)

<http://www.wireshark.org>

Using Wireshark



http-example.cap

Apply a display filter... < / >

| No. | All | Time | Source | Destination | Protocol | Info |
|-----|----------|---------------|---------------|-------------|---|------|
| 1 | 0.000000 | 172.24.65.102 | 91.102.91.18 | TCP | 58816 - http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TStamp=745562412 TSecr=0 SACK_PERM=0 | |
| 2 | 0.000170 | 172.24.65.102 | 91.102.91.18 | TCP | 58817 - http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TStamp=745562412 TSecr=0 SACK_PERM=0 | |
| 3 | 0.127853 | 91.102.91.18 | 172.24.65.102 | TCP | http - 58816 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=8 TStamp=18552... | |
| 4 | 0.127167 | 91.102.91.18 | 172.24.65.102 | TCP | http - 58817 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=8 TStamp=25124... | |
| 5 | 0.127181 | 172.24.65.102 | 91.102.91.18 | TCP | 58816 - http [ACK] Seq=1 Ack=1 Win=131760 Len=0 TStamp=745562538 TSecr=1855239975 | |
| 6 | 0.127226 | 172.24.65.102 | 91.102.91.18 | TCP | 58817 - http [ACK] Seq=1 Ack=1 Win=131760 Len=0 TStamp=745562538 TSecr=2512433875 | |
| 7 | 0.127363 | 172.24.65.102 | 91.102.91.18 | HTTP | GET / HTTP/1.1 | |
| 8 | 0.141328 | 91.102.91.18 | 172.24.65.102 | HTTP | HTTP/1.1 304 Not Modified | |
| 9 | 0.141421 | 172.24.65.102 | 91.102.91.18 | TCP | 58816 - http [ACK] Seq=503 Ack=100 Win=131568 Len=0 TStamp=745562551 TSecr=1855239975 | |

Frame 7: 568 bytes on wire (4544 bits), 568 bytes captured (4544 bits)
Ethernet II, Src: Apple_6c:87:5e (7c:d1:c3:6c:87:5e), Dst: Cisco_32:89:30 (44:2b:03:32:89:30)
Internet Protocol Version 4, Src: 172.24.65.102 (172.24.65.102), Dst: 91.102.91.18 (91.102.91.18)
Transmission Control Protocol, Src Port: 58816 (58816), Dst Port: http (80), Seq: 1, Ack: 1, Len: 502

HyperText Transfer Protocol
GET / HTTP/1.1
Host: 91.102.91.18
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.146 Safari/537.36
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8,cs;q=0.6,da;q=0.4
If-None-Match: "7053a63e1516a50b27a95edb31d07524a6e0a3"
If-Modified-Since: Tue, 17 Nov 2009 11:22:22 GMT

Full request URI: http://91.102.91.18/1
[HTTP request 1/1]
Response in frame: 81

0008 44 2b 02 32 09 30 7c d1 c3 6c 87 5e 00 45 00 D+..2.0!ñ Áí.~.E.
0018 02 2a 9e d7 40 00 40 06 f5 ff ac 18 41 66 5b 66 ..>..8g.~.Af!f
0028 5b 12 e5 c0 00 50 00 00 00 c7 03 14 0c 19 80 18 [..ñ.~.P.~.C.....
0038 20 2b 0f c0 00 00 02 01 08 00 2c 70 61 ae 66 94 .+.ñ....,.papn.
0040 b7 27 47 45 54 20 2f 20 48 54 50 2f 31 2e 31 .'GET / HTTP/1.1
0050 0d 0a 48 63 73 74 3a 20 39 31 2e 31 30 32 2e 39 ..Host: 91.102.9
0060 31 2e 31 30 0d 0a 43 60 6e 6a 65 63 74 69 6f 6e 1..18..Co nnection:
0070 2d 0a 43 60 6e 65 63 74 69 6f 6e 1..18..Co nnection:
0080 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 26 6d 61 78 che-Cont rol: max
0090 2d 61 67 65 3d 30 0d 0a 41 63 63 65 70 74 3a 20 ~age=0.. Accept:
00a0 74 65 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 text/html,application/
00b0 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c application/xhtml+xml,
00c0 61 70 70 6c 69 63 63 74 69 6f 6e 2f 78 6d 6c 3b application/xml;;

Packets: 9 - Displayed: 9 - Marked: 0 - Load time: 0:0:0 - Profile: Default

Capture - Options, select a network interface

Detailed view of network traffic with Wireshark



```
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 198
  ▼ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 194
    Version: TLS 1.2 (0x0303)
    ▶ Random
      Session ID Length: 0
      Cipher Suites Length: 32
    ▶ Cipher Suites (16 suites)
      Compression Methods Length: 1
    ▶ Compression Methods (1 method)
      Extensions Length: 121
    ▶ Extension: Unknown 56026
    ▶ Extension: renegotiation_info
    ▼ Extension: server_name
      Type: server_name (0x0000)
      Length: 16
      ▶ Server Name Indication extension
        Server Name list length: 14
        Server Name Type: host_name (0)
        Server Name length: 11
        Server Name: twitter.com
      ▶ Extension: Extended Master Secret
0050 a4 1d 52 8f 2c 18 99 91 54 68 0a 77 0d 95 73 64 ..R,... Th.w..sd
0060 7d 00 00 20 5a 5a c0 2b c0 2f c0 2c c0 30 cc a9 }.. ZZ.+ ./.,.0..
0070 cc a8 cc 14 cc 13 c0 13 c0 14 00 9c 00 9d 00 2f ..... ....../
0080 00 35 00 0a 01 00 00 79 da da 00 00 ff 01 00 01 .5.....y .....
0090 00 00 00 10 00 0e 00 00 0b 74 77 69 74 74 65 ..... ..twitte
00a0 72 2e 63 6f 6d 00 17 00 00 00 23 00 00 00 0d 00 r.com... ..#.....
00b0 14 00 12 04 03 08 04 04 01 05 03 08 05 05 01 08 ..... .....
```

Notice also the filtering possibilities, capture and view

Remote network debugging



- TShark and Tcpdump, I often use:
`tcpdump -nei eth0`
`tshark -z expert -r download-slow.pcapng`
- Remote packet dumps, `tcpdump -i eth0 -w packets.pcap`
- Story: tcpdump was originally written in 1988 by Van Jacobson, Sally Floyd, Vern Paxson and Steven McCanne who were, at the time, working in the Lawrence Berkeley Laboratory Network Research Group
<https://en.wikipedia.org/wiki/Tcpdump>

Great network security comes from knowing networks!

Note About Hardware IPv4 checksum offloading



IPv4 checksum must be calculated for every packet received

IPv4 checksum must be calculated for every packet sent

Usually on a router the Time To Live is decremented, to need re-calculation

Let an ASIC chip on the network card do the work!

Most server network chips today support this and more

Benefit for performance, but beware when using security tools

If every packet in wireshark has wrong checksum, its the network card doing it

Can be turned off, when doing security work

The basic tools for monitoring network



Moving from manual checking to automated

- Wireshark is advanced manual tool, try right-clicking different places
- How do we continuously monitor the network?

Automated packet sniffing tools



The Zeek Network Security Monitor

Zeek – Network Security Monitor <https://zeek.org>

Suricata – open source, mature, fast and robust network threat detection <https://suricata-ids.org/>
(Snort, old timer Intrusion Detection Engine (IDS))

Often a combination of tools and methods used in practice

Full packet capture big data tools also exist

The Zeek Network Security Monitor



The Zeek Network Security Monitor

Why Choose Zeek? Zeek is a powerful network analysis framework that is much different from the typical IDS you may know.

Adaptable

Zeek's domain-specific scripting language enables site-specific monitoring policies.

Efficient

Zeek targets high-performance networks and is used operationally at a variety of large sites.

Flexible

Zeek is not restricted to any particular detection approach and does not rely on traditional signatures.

Forensics

Zeek comprehensively logs what it sees and provides a high-level archive of a network's activity.

In-depth Analysis

Zeek comes with analyzers for many protocols, enabling high-level semantic analysis at the application layer.

Highly Stateful

Zeek keeps extensive application-layer state about the network it monitors.

Open Interfaces

Zeek interfaces with other applications for real-time exchange of information.

Open Source

Zeek comes with a BSD license, allowing for free use with virtually no restrictions.

The Zeek Network Security Monitor is not a single tool, more of a powerful network analysis framework

Zeek is the tool formerly known as Bro, changed name in 2018. <https://www.zeek.org/>

Zeek IDS is



The Zeek Network Security Monitor

While focusing on network security monitoring, Zeek provides a comprehensive platform for more general network traffic analysis as well. Well grounded in more than 15 years of research, Zeek has successfully bridged the traditional gap between academia and operations since its inception.

<https://www.Zeek.org/>

Zeek scripts



```
global dns_A_reply_count=0;
global dns_AAAA_reply_count=0;
...
event dns_A_reply(c: connection, msg: dns_msg, ans: dns_answer, a: addr)
{
    ++dns_A_reply_count;
}

event dns_AAAA_reply(c: connection, msg: dns_msg, ans: dns_answer, a: addr)
{
    ++dns_AAAA_reply_count;
}
```

source: dns-fire-count.bro from

<https://github.com/LiamRandall/bro-scripts/tree/master/fire-scripts>

<https://www.bro.org/sphinx-git/script-reference/scripts.html>

Side note: Zeek Security Monitor handles formats differently



Zeek has files formatted with a header:

```
#fields ts      uid      id.orig_h      id.orig_p      id.resp_h      id.resp_p      proto      trans_id
       rtt      query     qclass    qclass_name    qtype      qtype_name    rcode      rcode_name    AA
       TC       RD       RA        Z           answers    TTLs       rejected
```

```
1538982372.416180 CD12Dc1SpQm42QW4G3 10.xxx.0.145 57476 10.x.y.141 53 udp 20383
0.045021 www.dr.dk 1 C_INTERNET 1 A 0 NOERROR F F T T 0
  www.dr.dk-v1.edgekey.net,e16198.b.akamaiedge.net,2.17.212.93 60.000000,20409.000000,20.000000 F
```

Note: this show ALL the fields captured and dissected by Zeek, there is a nice utility program bro-cut which can select specific fields:

```
root@NMS-VM:/var/spool/bro/bro# cat dns.log | bro-cut -d ts query answers | grep dr.dk
2018-10-08T09:06:12+0200 www.dr.dk www.dr.dk-v1.edgekey.net,e16198.b.akamaiedge.net,2.17.212.93
```

Can also just use JSON now via Filebeat

Get Started with Zeek



To run in “base” mode: `bro -r traffic.pcap`

To run in a “near broctl” mode: `bro -r traffic.pcap local`

To add extra scripts: `bro -r traffic.pcap myscript.bro`

Note: the project was renamed from Bro to Zeek in Oct 2018 tools still named broctl etc.



Bro demo: Run

```
// install zeek first
kunoichi:~ root# broctl
Hint: Run the broctl "deploy" command to get started.
```

```
Welcome to BroControl 1.5
Type "help" for help.
```

```
[BroControl] > install
creating policy directories ...
installing site policies ...
generating standalone-layout.bro ...
generating local-networks.bro ...
generating broctl-config.bro ...
generating broctl-config.sh ...
...
```

```
kunoichi:etc root# grep eth0 node.cfg
interface=eth0
```

Zeek demo: Run Zeek



```
// back to Broctl and start it
[BroControl] > start
starting bro
// and then
kunoichi:bro root# pwd
/usr/local/var/spool/bro
kunoichi:bro root# tail -f dns.log
```

More examples at:

<https://www.bro.org/sphinx/script-reference/log-files.html>



DNS is important

Another tool that provides a basic SQL-frontend to PCAP-files

<https://www.dns-oarc.net/tools/packetq>

<https://github.com/DNS-OARC/PacketQ>

Going back in time and finding systems that visited a specific domain can explain when and where an infection started.

Deciding on which tool to use, Zeek or PacketQ depends on the situation.

Suricata IDS/IPS/NSM



Suricata is a high performance Network IDS, IPS and Network Security Monitoring engine.

<http://suricata-ids.org/> <http://openinfosecfoundation.org>

We will now move to the workshop materials:

Suricata, Zeek og DNS Capture

<https://github.com/kramse/security-courses/tree/master/courses/networking/suricatazeek-workshop>

Exercise at home – Your lab setup



- Go to GitHub, Find user Kramse, click through security-courses, courses, suricatazeek and download the PDF files for the slides and exercises:

<https://github.com/kramse/security-courses/tree/master/courses/networking/suricatazeek-workshop>

- Get the lab instructions, from

<https://github.com/kramse/kramse-labs/tree/master/suricatazeek>

NetFlow



NetFlow is getting more important, more data share the same links

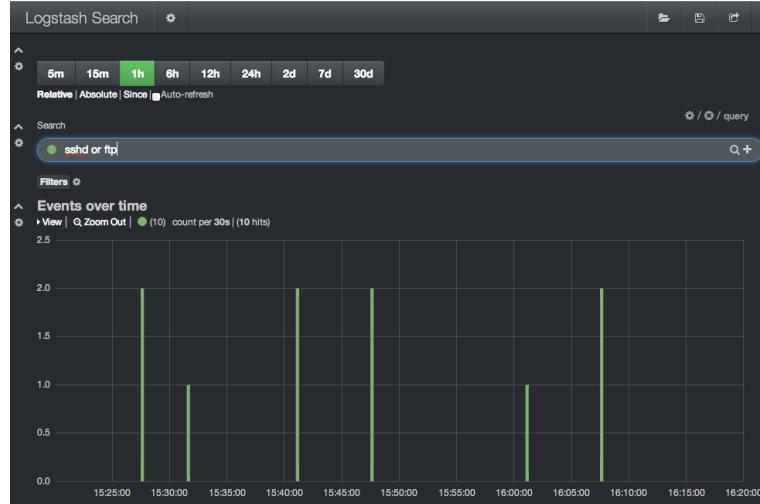
Accounting is important

Detecting DoS/DDoS and problems is essential

NetFlow sampling is vital information - 123Mbit, but what kind of traffic

Currently also investigating sFlow - hopefully more fine grained

View data efficiently

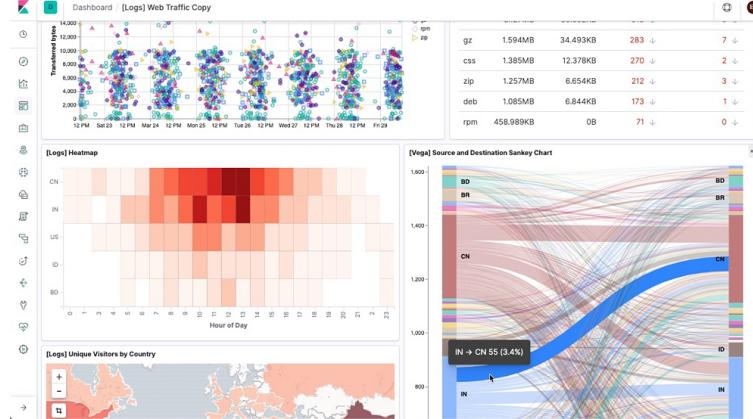


View data by digging into it easily - must be fast

Logstash and Kibana are just examples, but use indexing to make it fast!

Other popular examples include Graylog and Grafana

Big Data tools: Elasticsearch



Elasticsearch is an open source distributed, RESTful search and analytics engine capable of solving a growing number of use cases.

<https://www.elastic.co>

We are all Devops now, even security people!

Kibana

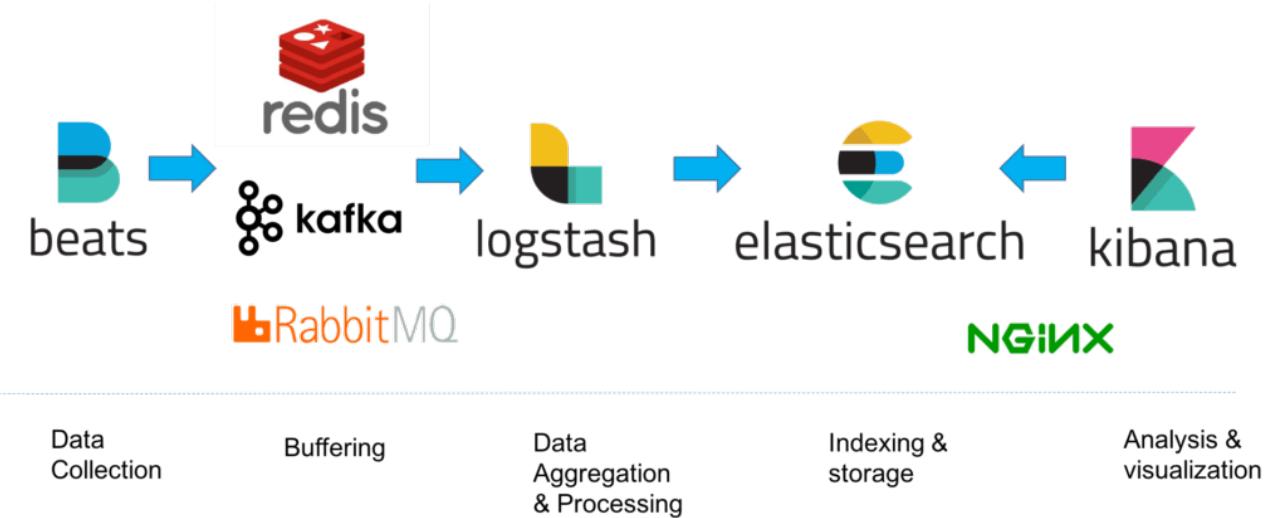


Highly recommended for a lot of data visualisation

Non-programmers can create, save, and share dashboards

Source: <https://www.elastic.co/products/kibana>

Architecture



- Real production environments often add some buffering in between
- Allows the ingestion to become more smooth, no lost messages

Metadata – enrichment



| Metadata | Metacategory: Data |
|---|--|
| The DNS lookup occurred at a certain time. | Timestamp: 278621182 |
| The internal host sent a DNS PTR request. | Network protocol: DNS PTR |
| The internal host had a hostname. | Location: Desktop subnet |
| | Source IP Address: 1.1.1.2 |
| | Hostname: windowspc22.company.com |
| The internal host resolved an external host. | Location: External |
| | Destination IP Address: 255.123.215.3 |
| | Hostname: dgf7adfnkjhh.com |
| The external host was hosted by a dynamic DNS provider. | Network: Shady DDnS Provider Inc. |
| | ASN: SHADY232 |
| | Reputation: Historically risky network |
| The remote hostname appeared randomly generated. | Hostname: dgf7adfnkjhh.com |
| | Category: Unusual, nonlinguistic |

Source: picture from *Crafting the InfoSec Playbook*, CIP

Metadata + Context

Summary, what to log



CIP 7 Tools of the Trade, need to know NetFlow, DNS, and HTTP proxy logs in the real-world

- Defense in Depth – we will never catch everything
- Log Management: The Security Event Data Warehouse
- Intrusion Detection Isn't Dead
- DNS, the One True King – Logging and analyzing DNS transactions, Blocking DNS requests or responses
- HTTP Is the Platform: Web Proxies – Web proxies allow you to solve additional security problems
- Rolling Packet Capture – In a perfect world, we would have full packet capture everywhere

Commercial Support



You can and should use updated rulesets for Suricata.

I Recommend the Emerging Threats ET Pro ruleset, which is about USD 900 per year per sensor. So two sites with Suricata running in both, 2x USD 900



Summary

We started from a basic Ubuntu/Debian server, and using Zeek and Suricata we now know more about our network.

We can collect logs about network traffic based on generic NetFlow, specific types DNS/TLS/protocols, and traffic matching advanced rulesets.

We know it is possible to create dashboards and visualizing the data.

What are the next steps?

Baseline Skills



- Threat-Centric Security, NSM, and the NSM Cycle
- TCP/IP Protocols
- Common Application Layer Protocols
- Packet Analysis
- Windows Architecture
- Linux Architecture
- Basic Data Parsing (BASH, Grep, SED, AWK, etc)
- IDS Usage (Snort, Suricata, etc.)
- Indicators of Compromise and IDS Signature Tuning
- Open Source Intelligence Gathering
- Basic Analytic Diagnostic Methods
- Basic Malware Analysis

Source: *Applied Network Security Monitoring Collection, Detection, and Analysis*, Chris Sanders and Jason Smith

Security devops



We need devops skillz in security

automate, security is also big data

integrate tools, transfer, sort, search, pattern matching, statistics, ...

tools, languages, databases, protocols, data formats

Use GitHub! So many libraries and programs that can help, maybe solve 90% of your problem, and you can glue the rest together

Example introductions:

- Seven languages/database/web frameworks in Seven Weeks
- Elasticsearch the definitive guide

We are all Devops now, even security people!

Questions?



Henrik Kramselund Jereminsen hkj@zecurity.com @kramse  

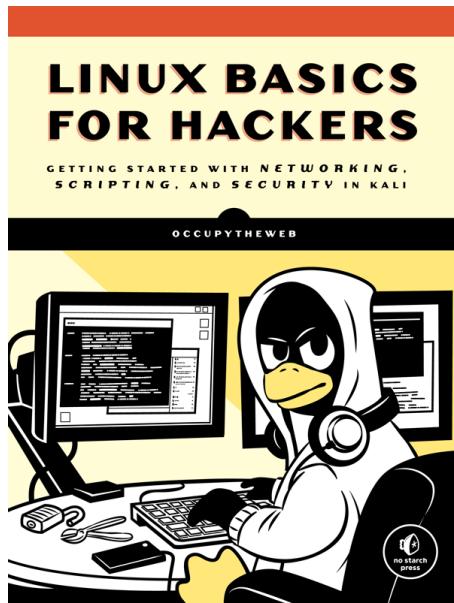
You are always welcome to send me questions later via email

Mobile: +45 2026 6000

Recommended further reading



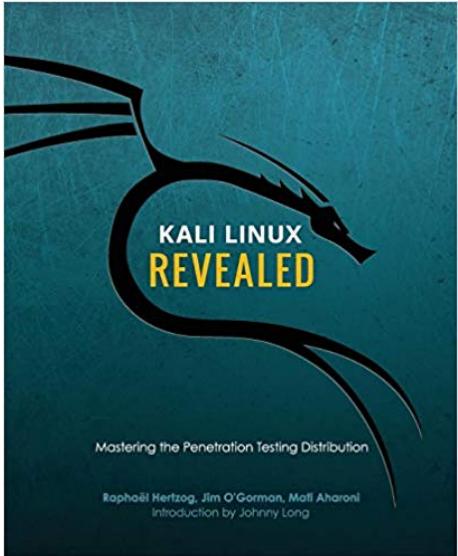
Book: Linux Basics for Hackers (LBhf)



Linux Basics for Hackers Getting Started with Networking, Scripting, and Security in Kali by OccupyTheWeb December 2018, 248 pp. ISBN-13: 9781593278557

<https://nostarch.com/linuxbasicsforhackers> Explains how to use Linux

Book: Kali Linux Revealed (KLR)



Kali Linux Revealed Mastering the Penetration Testing Distribution

<https://www.kali.org/download-kali-linux-revealed-book/>

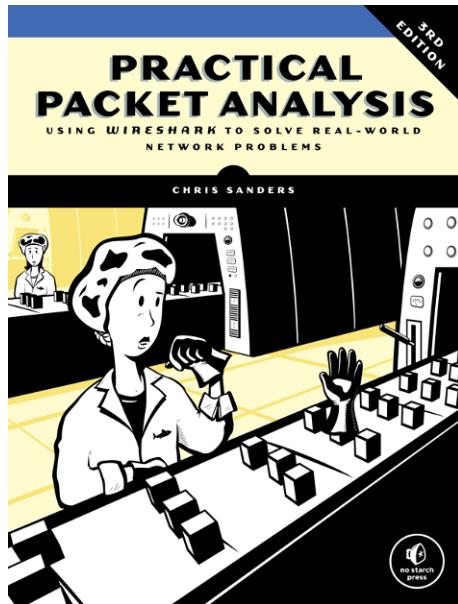
Explains how to install Kali Linux

Book: The Debian Administrator's Handbook (DEB)



The Debian Administrator's Handbook, Raphaël Hertzog and Roland Mas
<https://debian-handbook.info/> - shortened DEB

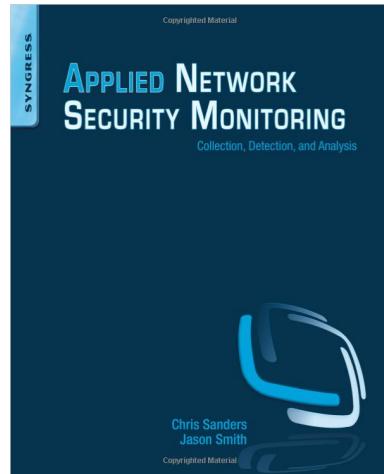
Book: Practical Packet Analysis (PPA)



Practical Packet Analysis, Using Wireshark to Solve Real-World Network Problems by Chris Sanders, 3rd Edition April 2017, 368 pp. ISBN-13: 978-1-59327-802-1

<https://nostarch.com/packetanalysis3>

Book: Applied Network Security Monitoring (ANSM)



Applied Network Security Monitoring: Collection, Detection, and Analysis 1st Edition

Chris Sanders, Jason Smith eBook ISBN: 9780124172166 Paperback ISBN: 9780124172081 496 pp. Imprint: Syngress, December 2013

<https://www.elsevier.com/books/applied-network-security-monitoring/unknown/978-0-12-417208-1>

Reading Summary



ANSM chapter (7,8),9,10 - 140 pages

DETECTION MECHANISMS

Generally, detection is a function of software that parses through collected data in order to generate alert data. This software is referred to as a detection mechanism.

Chapter 7 Detection Mechanisms, Indicators of Compromise, and Signatures

Chapter 8 Reputation-Based Detection

Chapter 9 Signature-Based Detection with Snort and Suricata

Chapter 10 The Bro Platform // Now Zeek

Zeek in the default configuration activates 10.000s of script lines out-of-the-box.

Gives great output with little effort and complements Suricata/NIDS

Reading Summary, continued



The Zeek Network Security Monitor

ANSM chapter 7: Detection Mechanisms and Indicators of Compromise, and Signatures

- Indicators of Compromise (IOC) any piece of information that can be used to objectively describe a network intrusion, expressed in a platform-independent manner
- Background information, useful when we talk about Zeek (previously Bro) later
- Intrusion Detection Systems try to detect ... but what if we know that some domains, servers, IPs etc are signs of bad activity - even compromise
- IP reputation - some IPs are used for controlling malware command and control (C2) servers etc.
- A signature can contain one or more IOCs

Reading Summary, continued



ANSM chapter 8: Reputation-Based Detection

- The most basic form of intrusion detection is reputation-based detection
- Similar concept to block lists for SMTP spam relays
- I often recommend <https://github.com/stamparm/maltrail> as a source of lists
- Other sources are lists like RIPE NCC delegated, which IP prefixes are handed out in different countries
<https://ftp.ripe.net/pub/stats/ripecc/delegated-ripecc-extended-latest>
ripecc|DK|ipv4|185.129.60.0|1024|20151130|allocated|
- Mentions SiLK <https://tools.netsa.cert.org/silk/>
If we end up having time today, or another day, we should look into this tool chain also!

Reading Summary, continued



ANSM chapter 9: Signature-Based Detection with Snort and Suricata

- Suricata IDS
- IDS rules are introduced
- I recommend a commercial license for the EmergingThreats ruleset

Reading Summary, continued



The Zeek Network Security Monitor

Why Choose Zeek? Zeek is a powerful network analysis framework that is much different from the typical IDS you may know.

Adaptable

Zeek's domain-specific scripting language enables site-specific monitoring policies.

Efficient

Zeek targets high-performance networks and is used operationally at a variety of large sites.

Flexible

Zeek is not restricted to any particular detection approach and does not rely on traditional signatures.

Forensics

Zeek comprehensively logs what it sees and provides a high-level archive of a network's activity.

In-depth Analysis

Zeek comes with analyzers for many protocols, enabling high-level semantic analysis at the application layer.

Highly Stateful

Zeek keeps extensive application-layer state about the network it monitors.

Open Interfaces

Zeek interfaces with other applications for real-time exchange of information.

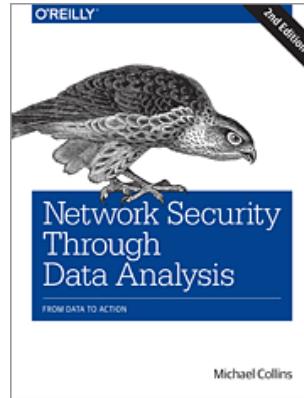
Open Source

Zeek comes with a BSD license, allowing for free use with virtually no restrictions.

ANSM chapter 10: The Zeek (Bro) Platform

- Zeek concepts and logs - many useful ones by default!

Network Security Through Data Analysis



Low page count, but high value! Recommended.

Network Security through Data Analysis, 2nd Edition By Michael S Collins Publisher: O'Reilly Media 2015-05-01:
Second release, 348 Pages

New Release Date: August 2017