Welcome to

1. Initial Overview of SIEM Terms

KEA Kompetence SIEM and Log Analysis

Henrik Kramselund he/him han/ham hlk@zencurity.com @kramse

Slides are available as PDF, kramse@Github

1-overview-of-siem-terms.tex in the repo security-courses

Goals for today

Todays goals:

- Introduce some terms and technologies
- Make it possible to run Python – easy on Debian
- Learn to find resources, data files and programs/libraries
- Put SIEM into context of the information security picture

Photo by Thomas Galler on Unsplash

Plan for today

Subjects

- Introduction to R, Python, using the book
- Github/Git
- SIEM and SOC
- Context, what are the threats, what are the answers we want from the SIEM and Logs

Exercise theme: Building a house requires materials

- Python Libraries - git clone
- Date formats ISO 8601
- Grok patterns, regular expressions
- Fetching resources from the internet – http-resources

Time schedule

- 17:00 - 18:15
- 30min break
- 18:45 - 19:30
- 15min break
- 19:45 -20:30 45min

Reading Summary

DDS 1. The Journey to Data-Driven Security

DDS 2. Building Your Analytics Toolbox: R and Python

CIP 1 Incident Response Fundamentals

CIP 2 What Are You Trying to Protect?

Skim CIP 3 What Are the Threats?

Reading Summary, continued

> Data-Driven Security: Analysis, Visualization and Dashboards has been designed to take you on a journey into the world of security data science. The start of the journey looks a bit like the word cloud shown in Figure 1, which was created from the text in the chapters of this book. You have a great deal of information available to you, and may be able to pick out a signal or two within the somewhat hazy noise on your own.
> However, it's like looking for a needle in a haystack without a magnet.

Source: DDS 1. The Journey to Data-Driven Security

- A history of Learning from Data
- Data Analysis Skills
- Exploring Data
- Tools needed vary, I recommend the editor Atom, install using Ansible ☺

Reading Summary, continued

> A discussion of which programming language is better than another for a certain set of tasks often turns (quickly) into a religious war of words that rarely wins converts and never becomes fully resolved. As a security data scientist, you will find that you do not have the luxury of language bias. There will be times when one language shines in one area while a different one shines in another, and you need the skills of a diplomat to bring them both together to solve real problems.

Source: DDS 2. Building Your Analytics Toolbox: R and Python

- Some tools are needed, some might be nice to know
- Python is a need to know
- R is a nice to know

Reading Summary, continued

- Keeping an organization safe from attack, as well as having a talented team available to respond quickly, minimizes damage to your reputation and business.
- Fostering and developing relationships with IT, HR, legal, executives, and others is critical to the success of a CSIRT.
- Sharing incident and threat data with external groups improves everyone's security and gives your organization credibility and trust with groups that might be able to help in the future.
- A good team relies on good tools, and a great team optimizes their operations.
- A solid and well-socialized InfoSec policy gives the incident response team the authority and charter to protect networks and data.

Source: CIP 1 Incident Response Fundamentals

Reading Summary, continued

- You can't properly protect your network if you don't know what to protect.
- Define and understand your critical assets and what's most important to your organization.
- Ensure that you can attribute ownership or responsibility for all systems on your network.
- Understand and leverage the log data that can help you determine host owner- ship.
- A complex network is difficult to protect, unless you understand it well.

Source: CIP 2 What Are You Trying to Protect?

Reading Summary, continued

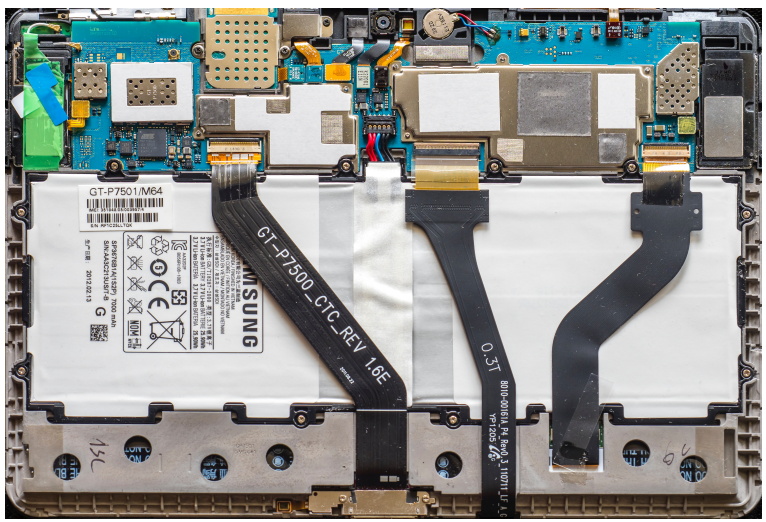
- To protect your organization, you must understand the threats it faces.
- If you don't think you have something to lose, you haven't thought about it enough.
- Crime evolves with culture and society. Online crime will increase as more things of value are digitally stored and globally accessible.
- Malicious activity can come from a number of sources, but the most common source is organized crime, followed by targeted attackers and trusted insiders.
- Different organizations face different threats; focus your efforts on protecting the high-value assets and make sure you monitor them closely.

Source: CIP 3 What Are the Threats?

A history of Learning from Data

- All our books contain a lot of example data
- Often you start from a few sources, to prove the worth

What is Infrastructure



- Enterprises today have a lot of computing systems supporting the business needs
- These are very diverse and often discrete systems

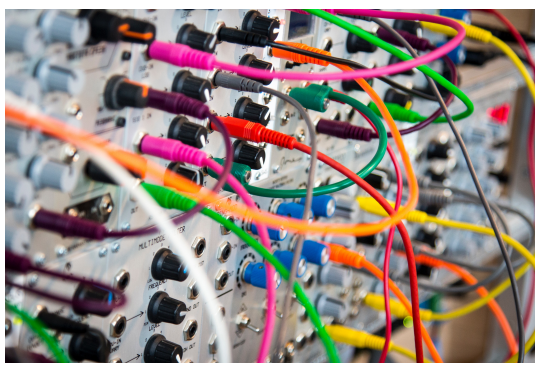
Photo by Alexander Schimmeck on Unsplash

Business Challenges

- Accumulation of software
- Legacy systems
- Partners
- Various types of data
- Employee churn, replacement

Photo by Adam Bignell on Unsplash

Software Challenges



- Complexity
- Various languages
- Various programming paradigms, client server, monolith, Model View Controller
- Conflicting data types and available structures
- Steam train vs electric train

Photo by John Barkiple on Unsplash

Developers Challenges

- Work in teams across organisation - and partners, vendors, sub-contractors
- Work with legacy systems, old technology
- Learn new Technologies

Photo by Kelly Sikkema on Unsplash

Integration Challenges

- Enable communication between components
- Need mediator, interpreter, translator
- Recognize standard patterns

Photo by Thomas Drouault on Unsplash

SIEM

Security information and event management (SIEM) is a subsection within the field of computer security, where software products and services combine security information management (SIM) and security event management (SEM). They provide real-time analysis of security alerts generated by applications and network hardware.

Vendors sell SIEM as software, as appliances, or as managed services; these products are also used to log security data and generate reports for compliance purposes.[1]

The term and the initialism SIEM was coined by Mark Nicolett and Amrit Williams of Gartner in 2005.[2]

Source: https://en.wikipedia.org/wiki/Security_information_and_event_management

- Note: there are alerting examples towards the bottom of the page, with sources
- Closely related to log management, incident response

SOC

> An information security operations center (ISOC or SOC) is a facility where enterprise information systems (web sites, applications, databases, data centers and servers, networks, desktops and other endpoints) are monitored, assessed, and defended.
>
> ...
>
> A security operations center (SOC) can also be called a security defense center (SDC), security analytics center (SAC), network security operations center (NSOC),[3] security intelligence center, cyber security center, threat defense center, security intelligence and operations center (SIOC). In the Canadian Federal Government the term, infrastructure protection center (IPC), is used to describe a SOC.

Source: https://en.wikipedia.org/wiki/Information_security_operations_center

- We have a whole book about SOCs, but I skipped the introductory chapters!
- If you need to build a SOC, that is great source of information

Subjects:

Context, what are the threats, what are the answers we want from the SIEM and Logs

What are the common cases, where we use the data?

- Incident Response
- Computer Emergency Response Team (CERT) and Computer Security Incident Response Teams (CSIRT)
- Security Departments
- GDPR Data protection
- Computer Forensics

Incident Handling, phases

The procedures developed for incident response must cover the complete life-cycle

- Preparation for an attack, establish procedures and mechanisms for detecting and responding to attacks
- Identification of an attack, notice the attack is ongoing
- Containment (confinement) of the attack, limit effects of the attack as much as possible
- Eradication of the attack, stop attacker, block further similar attacks
- Recovery from the attack, restore system to a secure state
- Follow-up to the attack, include lessons learned improve environment

Crafting the InfoSec Playbook

This book will help you to answer common questions:

- How do I find bad actors on my network?
- How do I find persistent attackers?
- How can I deal with the pervasive malware threat?
- How do I detect system compromises?
- How do I find an owner or responsible parties for systems under my protection?
- How can I practically use and develop threat intelligence?
- How can I possibly manage all my log data from all my systems?
- How will I benefit from increased logging—and not drown in all the noise?
- How can I use metadata for detection?

Source: *Crafting the InfoSec Playbook: Security Monitoring and Incident Response Master Plan* by Jeff Bollinger, Brandon Enright, and Matthew Valites ISBN: 9781491949405

MITRE ATT&CK framework

MITRE ATT&CK™ is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

With the creation of ATT&CK, MITRE is fulfilling its mission to solve problems for a safer world — by bringing communities together to develop more effective cybersecurity. ATT&CK is open and available to any person or organization for use at no charge.

ATT&CK™

Source: https://attack.mitre.org/ Great resource for attack categorization

Incident Response Checklists

Table 3-5. Incident Handling Checklist

	Action	Completed
	Detection and Analysis	
1.	Determine whether an incident has occurred	
1.1	Analyze the precursors and indicators	
1.2	Look for correlating information	
1.3	Perform research (e.g., search engines, knowledge base)	
1.4	As soon as the handler believes an incident has occurred, begin documenting the investigation and gathering evidence	
2.	Prioritize handling the incident based on the relevant factors (functional impact, information impact, recoverability effort, etc.)	
3.	Report the incident to the appropriate internal personnel and external organizations	
	Containment, Eradication, and Recovery	
4.	Acquire, preserve, secure, and document evidence	
5.	Contain the incident	
6.	Eradicate the incident	
6.1	Identify and mitigate all vulnerabilities that were exploited	
6.2	Remove malware, inappropriate materials, and other components	
6.3	If more affected hosts are discovered (e.g., new malware infections), repeat the Detection and Analysis steps (1.1, 1.2) to identify all other affected hosts, then contain (5) and eradicate (6) the incident for them	
7.	Recover from the incident	
7.1	Return affected systems to an operationally ready state	
7.2	Confirm that the affected systems are functioning normally	
7.3	If necessary, implement additional monitoring to look for future related activity	
	Post-Incident Activity	
8.	Create a follow-up report	
9.	Hold a lessons learned meeting (mandatory for major incidents, optional otherwise)	

This checklist is from the NIST document *Computer Security Incident Handling Guide: Recommendations of the National Institute of Standards and Technology*, NIST Special Publication 800-61 Revision 2, August 2012.

CIS Controls also recommend Incident Response

CIS Control 19:

Incident Response and Management Protect the organization's information, as well as its reputation, by developing and implementing an incident response infrastructure (e.g., plans, defined roles, training, communications, management oversight) for quickly discovering an attack and then effectively containing the damage, eradicating the attacker's presence, and restoring the integrity of the network and systems.

Source: Center for Internet Security CIS Controls 7.1 CIS-Controls-Version-7-1.pdf from https://www.cisecurity.org/controls/

Anatomy of an Auditing System

Sample logs from login with Secure Shell (SSH) and performing `sudo su -`

```
Jun  5 11:53:15 pumba sshd[64505]: Accepted publickey for hlk from 79.142.233.18 port 43902
 ssh2: ED25519 SHA256:l8OJMcywyBcraJiCWJ06uZ2yzHfu0VuiArqVvlVyfEI

Jun  5 11:53:19 pumba sudo:      hlk : TTY=ttyp2 ; PWD=/home/hlk ; USER=root ; COMMAND=/usr/bin/su -
```

Example systems: Unix syslog, IBM main frame RACF and Windows Event Logs service

swatchdog is an old skool, but simple tool that works

Logs should be protected and considered confidential information

Anatomy of an Auditing System

When data has been gathered it should be analyzed.

Logger functions - collect

Analyzer - analyze it, creating dashboard can provide some insights

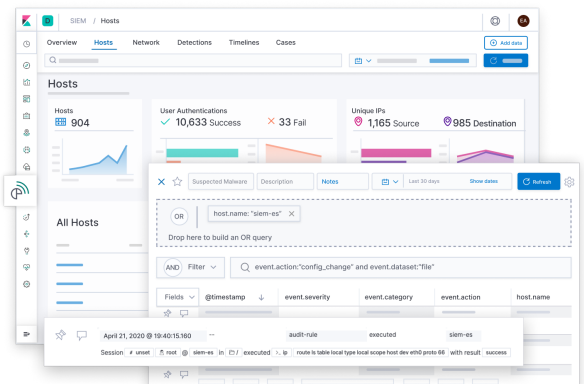
Notifier - report results by email or other means

Example systems Windows Event Logs service can inform of successful and failed logins, both should be collected

Logs should be protected and considered confidential information, by sending it to a centralized system with a high security level protects it

Modern systems exist to take all data from logging and provide high capacity storage, searching and sorting.

Why Elasticsearch



Screenshot from https://www.elastic.co/siem

Recommend building a proof-of-concept infrastructure using the Elastic stack and gather experience with logging. This can be done without a license fee and the organization can then see what works and doesn't. Then using the experiences as input an informed decision can be made, to continue with this as a home grown logging and auditing solution, or buy a premade one.

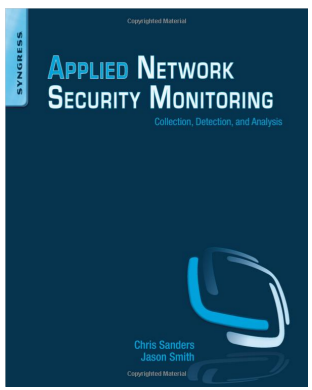
Technologies used in this course

The following tools and environments are examples that may be introduced in this course:

- Programming languages and frameworks Java, Python, regular expressions
- Development environments – choose your own IDE / Editor – I use **Atom**
- Networking and network protocols: TCP/IP, HTTP, DNS, Netflow
- Formats XML, JSON, CSV, raw text, web scraping
- Web technologies and services: REST, API, HTML5, CSS, JavaScript
- Tools like cURL, Zeek, Git and Github
- Message queueing systems: MQ and Redis could be added
- Aggregated example platforms: Elastic stack, logstash, elasticsearch, kibana, grafana, Filebeat
- Cloud and virtualisation Docker, Kubernetes, Azure, AWS, microservices – can be added

This list is not complete or a promise

Book: Applied Network Security Monitoring (ANSM)



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

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

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

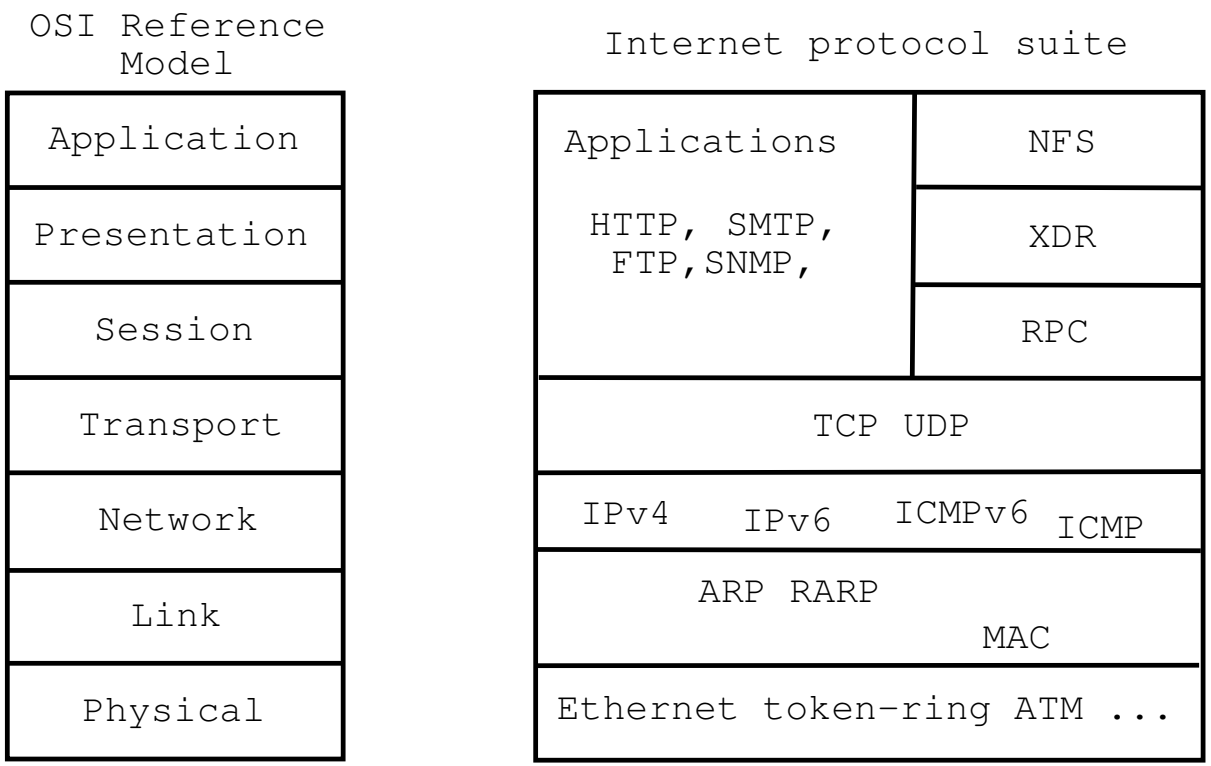
We use this in the course Network and Communication Security at KEA

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

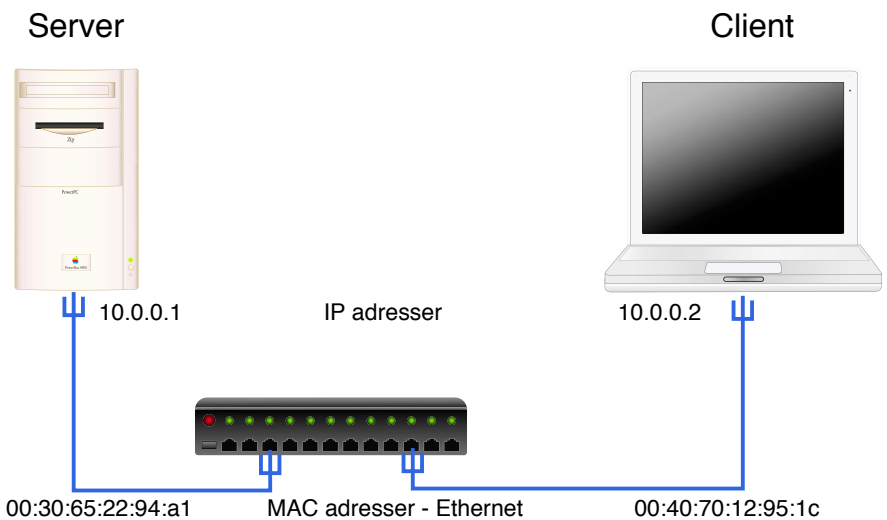
Source: *Applied Network Security Monitoring Collection, Detection, and Analysis*, Chris Sanders and Jason Smith

OSI and Internet



Data on all layers

Networking in TCP/IP



- Everything uses TCP/IP today, more or less.
- Clients make requests, receives responses
- HyperText Transfer Protocol (HTTP) is an example

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**.

Strategy for implementing identification and detection

We recommend that the following strategy is used for implementing identification and detection.

We have the following recommendations and actions points for logging:

- ☐ Enable system logging from servers
- ☐ Enable system logging from network devices
- ☐ Centralize logging
- ☐ Add search facilities and dashboards
- ☐ Perform system audits manually or automatically
- ☐ Setup notification and notification procedures

Extended Sources

When a basic logging infrastructure is setup, it can be expanded to increase coverage, by adding more sources:

- DNS query logging – will enable multiple cases to be resolved, example malware identification and tracing, when was a malware domain queried, when was the first infection
- Session data from Firewalls, Netflow – traffic patterns can be investigated and both attacks and cases like exfiltration can likely be seen

Hint: Take the sources available first, make a proof-of-concept, expand coverage

Data Analysis Skills

Although we could spend an entire book creating an exhaustive list of skills needed to be a good security data scientist, this chapter covers the following skills/domains that a data scientist will benefit from knowing within information security:

- Domain expertise—Setting and maintaining a purpose to the analysis
- Data management—Being able to prepare, store, and maintain data
- Programming—The glue that connects data to analysis
- Statistics—To learn from the data
- Visualization—Communicating the results effectively

It might be easy to label any one of these skills as the most important, but in reality, the whole is greater than the sum of its parts. Each of these contributes a significant and important piece to the workings of security data science.

Source: *Data-Driven Security: Analysis, Visualization and Dashboards* Jay Jacobs, Bob Rudis
ISBN: 978-1-118-79372-5 February 2014 https://datadrivensecurity.info/ - short DDS

Don't use spreadsheets!

- Spreadsheets are great for some tasks, but ...
- They don't scale
- The model can be broken – edit a single formula
- Rounding errors accumulate
- Input and output are limited
- Most functions require manual work

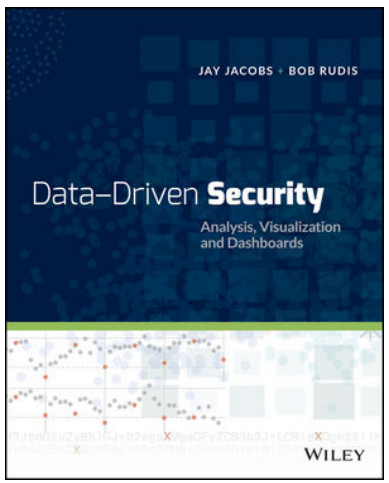
Start programming

Example program, reading from HTTP, into Python, out using JSON

```
#!/usr/bin/env python
import requests
r = requests.get('https://api.github.com/events')
print (r.json());
```

- Think of programing in this course as as reading, processing and outputting data
- Read in any format
- Process with any function
- Output in any format
- And reuse existing software

DDS: Chapter 2 exercises



- Lets get R and Python running
- Try as many of the scripts and examples as we want

Data overview XML data, JSON

Photo by Chris Lawton on Unsplash

XML data

> Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The World Wide Web Consortium's XML 1.0 Specification[2] of 1998[3] and several other related specifications[4]—all of them free open standards—define XML.[5]
>
> The design goals of XML emphasize simplicity, generality, and usability across the Internet.[6] It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures[7] such as those used in web services.

Source: https://en.wikipedia.org/wiki/XML

- We have seen XML used for configuration in Apache Tomcat and Camel
- Perfect for computers, less for humans

XML data example - Nmap output

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 7.70 scan initiated Sat Feb 22 23:35:53 2020 as: nmap -oA router -sP 10.0.42.1 -->
<nmaprun scanner="nmap" args="nmap -oA router -sP 10.0.42.1" start="1582410953"
 startstr="Sat Feb 22 23:35:53 2020" version="7.70" xmloutputversion="1.04">
<verbose level="0"/>
<debugging level="0"/>
<host><status state="up" reason="echo-reply" reason_ttl="62"/>
<address addr="10.0.42.1" addrtype="ipv4"/>
<hostnames>
</hostnames>
<times srtt="2235" rttvar="5000" to="100000"/>
</host>
<runstats><finished time="1582410953" timestr="Sat Feb 22 23:35:53 2020" elapsed="0.32"
 summary="Nmap done at Sat Feb 22 23:35:53 2020; 1 IP address (1 host up)
 scanned in 0.32 seconds" exit="success"/><hosts up="1" down="0" total="1"/>
</runstats>
</nmaprun>
```

XML data - documents

> Hundreds of document formats using XML syntax have been developed,[8] including RSS, Atom, SOAP, SVG, and XHTML. XML-based formats have become the default for many office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org and LibreOffice (OpenDocument), and Apple's iWork[citation needed]. XML has also provided the base language for communication protocols such as XMPP. Applications for the Microsoft .NET Framework use XML files for configuration, and property lists are an implementation of configuration storage built on XML.[9]

Source: https://en.wikipedia.org/wiki/XML

- Document formats using XML may still be proprietary!
- Documents using XML can be validated, are they well-formed according to the Document Type Definition (DTD)

Transforming XML using XSLT

> XSLT (Extensible Stylesheet Language Transformations) is a language for transforming XML documents into other XML documents,[1] or other formats such as HTML for web pages, plain text or XSL Formatting Objects, which may subsequently be converted to other formats, such as PDF, PostScript and PNG.[2] XSLT 1.0 is widely supported in modern web browsers.[3]

Source: https://en.wikipedia.org/wiki/XSLT

- Can be seen as a mapping between formats, different XML schemas
- Also is Turing complete, is a programming language

XSLT example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="/persons">
    <root> <xsl:apply-templates select="person"/> </root>
  </xsl:template>
  <xsl:template match="person">
    <name username="{@username}"> <xsl:value-of select="name" /> </name>
  </xsl:template>
</xsl:stylesheet>
```

- XSLT uses XPath to identify subsets of the source document tree and perform calculations. XPath also provides a range of functions
- XSLT functionalities overlap with those of XQuery, which was initially conceived as a query language for large collections of XML documents
  Source: https://en.wikipedia.org/wiki/XSLT

xsltproc example using Nmap

```
$ su -
# apt install nmap xsltproc
# nmap -sP -oA /tmp/router 91.102.91.18
# exit
$ xsltproc /tmp/router.xml > /tmp/router.html
$ firefox /tmp/router.html
```

- We can use the command line tool `xlstproc` for transforming documents
- `apt install xsltproc`
- Its part of the package Libxslt https://en.wikipedia.org/wiki/Libxslt

- Try installing the tools Nmap and `xsltproc` and reproduce the above
- This is an easy tool to try, and very useful too

Data overview JSON

> JavaScript Object Notation (JSON, pronounced / d e sən/; also / d e s n/[note 1]) is an open-standard file format or data interchange format that uses **human-readable text** to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format, with a diverse range of applications, such as serving as replacement for XML in AJAX systems.[6]

Source: https://en.wikipedia.org/wiki/JSON

- I like JSON much better than XML
- Many web services can supply data in JSON format

JSON example

```
{
  "first name": "John",
  "last name": "Smith",
  "age": 25,
  "address": {
    "street address": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postal code": "10021"
  },
  "phone numbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
  ],
}
```

- This is a basic JSON document, new data attribute-value pairs can be added
  Source: https://en.wikipedia.org/wiki/JSON

Python and REST

```
#!/usr/bin/env python
import requests
r = requests.get('https://api.github.com/events')
print (r.json());
```

- Lets try to use some Python to access a REST service.
- We will use the JSONPlaceholder which is a free online REST API: https://jsonplaceholder.typicode.com/
- Start at the site: https://jsonplaceholder.typicode.com/guide.html and try running a few of the examples with your browser
- Then try using the same URLS in the Requests HTTP library from Python, https://requests.readthedocs.io/en/master/

Note about frameworks and libraries

```
import xml.etree.ElementTree as ET
tree = ET.parse('testfile.xml')
root = tree.getroot()

print(root.tag)
print('Nmap version: \t\t{:s} '.format(root.attrib['version']))
print('Nmap started: \t\t{:s} '.format(root.attrib['startstr']))
print('Nmap command line: \t{:s} '.format(root.attrib['args']))

hosts = tree.findall('./host')
for host in hosts:
    print(host.tag)
    print(host.attrib)
    for hostvalues in host:
        print(hostvalues.tag)
        print(hostvalues.attrib)
```

- Dont import JSON or XML using home made programs
- Example uses xml.etree.ElementTree from Python https://docs.python.org/3.7/library/xml.etree.elementtree.html

Convert XML to JSON

```
import xml.etree.ElementTree as ET
import json
def etree_to_dict(t):
    d = {t.tag : map(etree_to_dict, t.getchildren())}
    d.update(('@' + k, v) for k, v in t.attrib.items())
    d['text'] = t.text
    return d

tree = ET.parse('testfile.xml')
root = tree.getroot()
mydict = etree_to_dict(root)
print(type(tree))
print(type(root))
print(type(mydict))

print(mydict)

with open('testfile.json', 'w') as json_file:
  json.dump(mydict, json_file)
```

Converting using Python is easy

Side note: Zeek Security Monitor handles formats differently

Zeek has files formatted with a header:

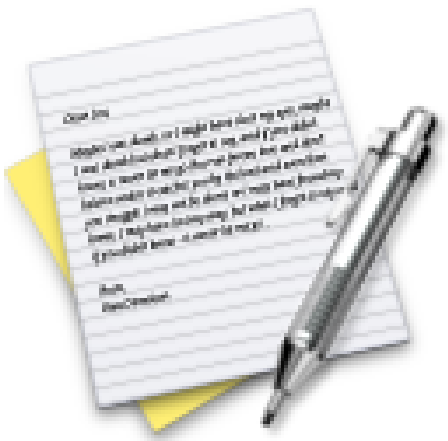
```
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   trans_id
        rtt     query   qclass  qclass_name     qtype   qtype_name      rcode   rcode_name      AA
        TC      RD      RA      Z       answers TTLs    rejected

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 zeek-cut which can select specific fields:

```
root@NMS-VM:/var/spool/bro/bro# cat dns.log | zeek-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

Exercise



Now lets do the exercise

Use a XML library in Python – up to 30min

which is number 11 in the exercise PDF.

Functions found on the internet, or in libraries

- Probably every function we require in this course is already implemented
- Reuse libraries
- Import math libraries to get statistics
- Parsing of structures – import parsing library
- Need reading a file format, find the package for it
- Even machine learning – import it
- Visualize using a framework

Git intro

> Git (/ t/)[7] is a distributed version-control system for tracking changes in source code during software development.[8] It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed,[9] data integrity,[10] and support for distributed, non-linear workflows.[11]

Source: https://en.wikipedia.org/wiki/Git

- We will introduce Git and Github - commercial Git hosting company https://en.wikipedia.org/wiki/Git
- Try creating a Git repository, remember to add a README
- Checkout the repository
- Edit a file
- add and commit it

Use the Github Hello World example: https://guides.github.com/activities/hello-world/

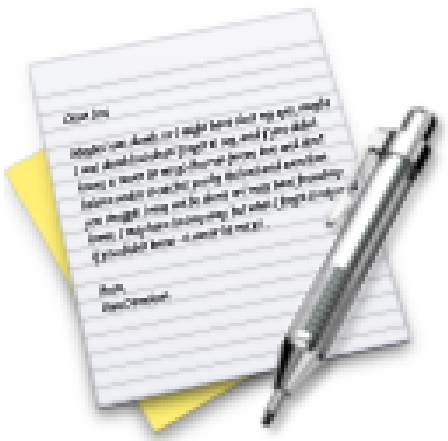
My daily job – Security engineering a job role

On any given day, you may be challenged to:

- Create new ways to solve existing production security issues
- Configure and install firewalls and intrusion detection systems
- Perform vulnerability testing, risk analyses and security assessments
- Develop automation scripts to handle and track incidents
- Investigate intrusion incidents, conduct forensic investigations and incident responses
- Collaborate with colleagues on authentication, authorization and encryption solutions
- Evaluate new technologies and processes that enhance security capabilities
- Test security solutions using industry standard analysis criteria
- Deliver technical reports and formal papers on test findings
- Respond to information security issues during each stage of a project's lifecycle
- Supervise changes in software, hardware, facilities, telecommunications and user needs
- Define, implement and maintain corporate security policies
- Analyze and advise on new security technologies and program conformance
- Recommend modifications in legal, technical and regulatory areas that affect IT security

Source: https://www.cyberdegrees.org/jobs/security-engineer/
also https://en.wikipedia.org/wiki/Security_engineering

Exercise



Now lets do the exercise

Clone a Python library StevenBlack/hosts – 30min

which is number 12 in the exercise PDF.

A warning about dates!

- Remember September 1752
- Dates can be tricky
- Use a standard date format ISO 8601
- *Falsehoods programmers believe about time*
  https://infiniteundo.com/post/25326999628/falsehoods-programmers-believe-about-time
- Updated with more:
  https://infiniteundo.com/post/25509354022/more-falsehoods-programmers-believe-about-time

Exercise

Now lets do the exercise

⚠ Date Formats – 15min

which is number **13** in the exercise PDF.

Grok expresssions

```
filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp}
      %{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}
      (?:\[%{POSINT:syslog_pid}\])?: %{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    syslog_pri { }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}
```

- Logstash filter expressions grok can normalize and split data into fields

Source: Config snippet from recommended link

http://logstash.net/docs/1.4.1/tutorials/getting-started-with-logstash

Grok expresssions, sample from my archive

```
filter {
# decode some SSHD
if [syslog_program] == "sshd" {
  grok {
# May 20 10:27:08 odn1-nsm-01 sshd[4554]: Accepted publickey for hlk from
10.50.11.17 port 50365 ssh2: DSA 9e:fd:3b:3d:fc:11:0e:b9:bd:22:71:a9:36:d8:06:c7

match => { "message" => "%{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target}
sshd\[%{BASE10NUM}\]: Accepted publickey for %{USERNAME:username} from
  %{IP:src_ip} port %{BASE10NUM:port} ssh2" }

# "May 20 10:27:08 odn1-nsm-01 sshd[4554]: pam_unix(sshd:session):
session opened for user hlk by (uid=0)"
match => { "message" => "%{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host_target}
sshd\[%{BASE10NUM}\]: pam_unix\(sshd:session\): session opened for user
%{USERNAME:username}" }
```

- Logstash filter expressions grok can normalize and split data into fields

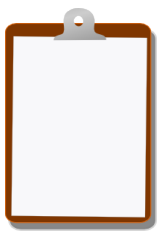
Exercise

Now lets do the exercise

Grok Debugger – 30min

which is number **14** in the exercise PDF.

For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools