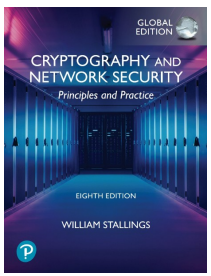
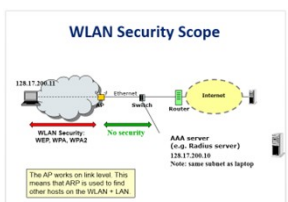
Course summary

Material included

- The book



- The slides

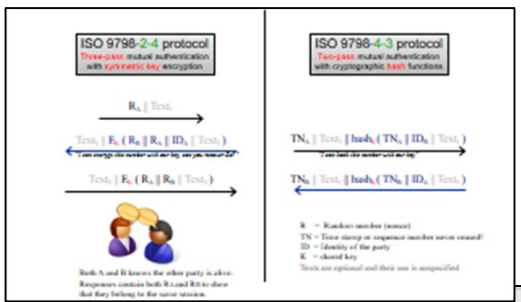


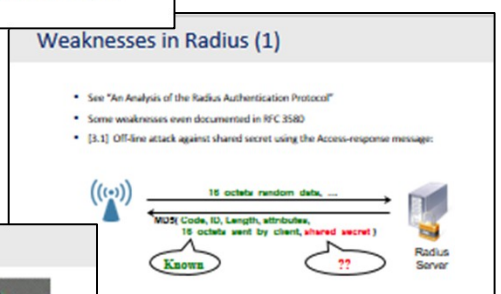
- The mandatory reading material

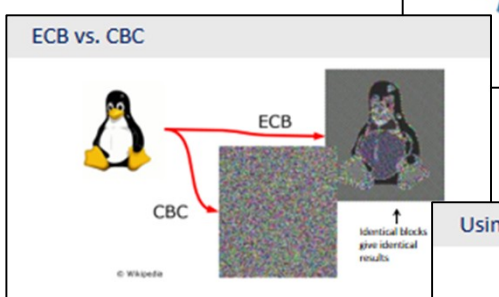
- All lab-related material

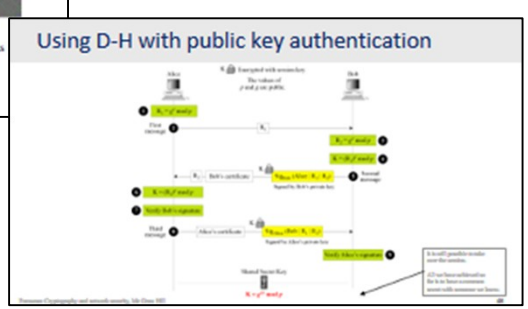
Topics covered

- Authentication, different methods
  - Radius
  - CHAP, ISO 9798
  - Certificates
- Cryptographic tools
  - Block and stream ciphers
  - MAC, HMAC
  - Diffie-Hellman
  - RSA



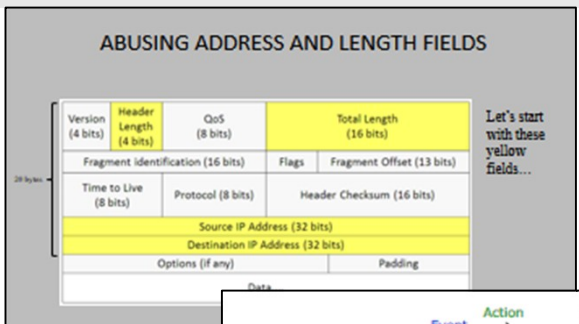


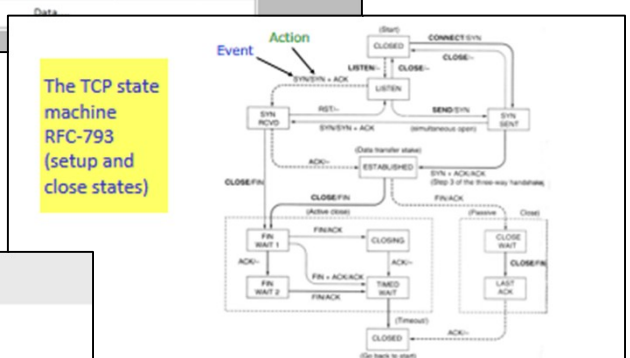


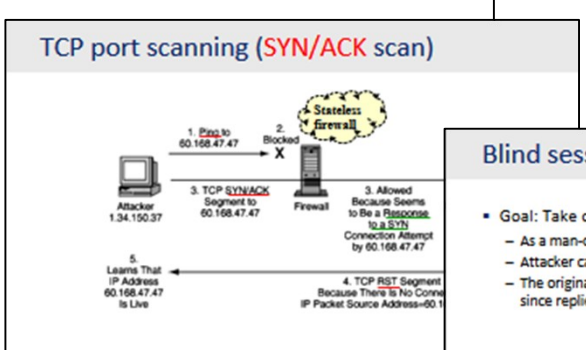


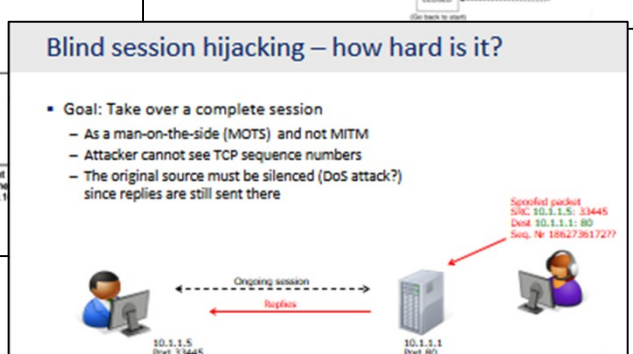
Topics covered

- Network layer security (IP)
  - Scanning, fragmentation, header abuse, address spoofing, ...
  - LAND attack (own address and port), Teardrop attack (fragmentation), …
- Transport layer security
  - Scanning hosts and ports
  - OS fingerprinting
  - DoS and DDoS attacks
  - Spoofing TCP segments and Sequence number prediction (blind injection), …



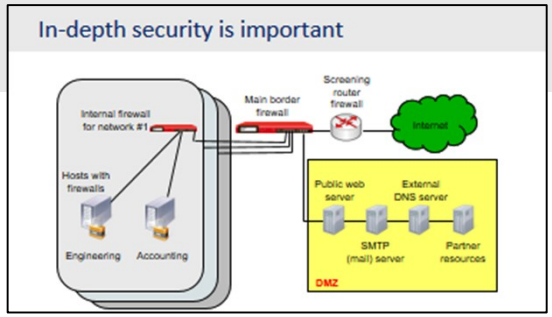


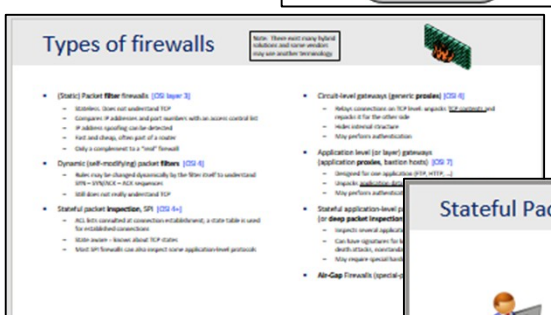


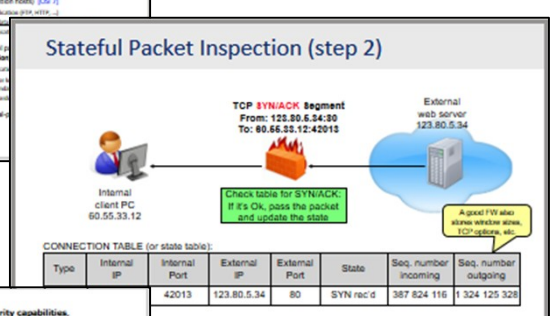


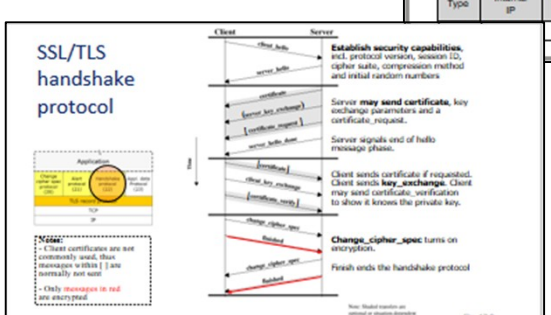
Topics covered

- Firewalls
  - Screening routers
  - Types of firewalls
  - Rules: ingress and egress filtering
  - State tables
- NAT – Network address translation
- IDS systems, Snort
- Evading FW and IDS
- SSL/TLS
  - Key components of a security protocol
  - Detailed protocol analysis
  - Changes in version 1.3
  - …



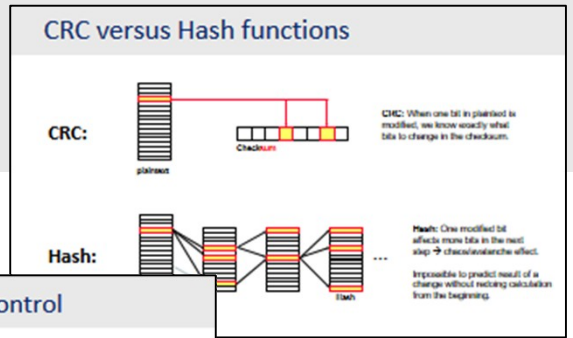


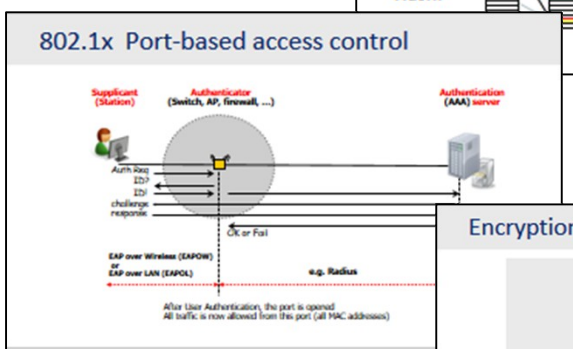


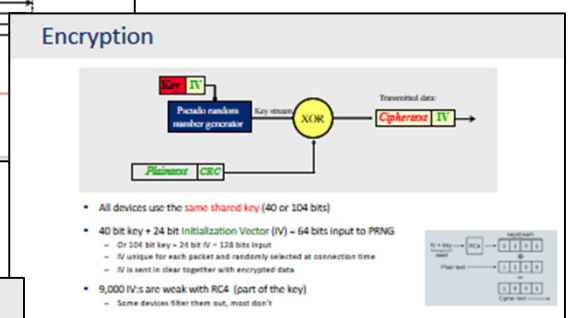


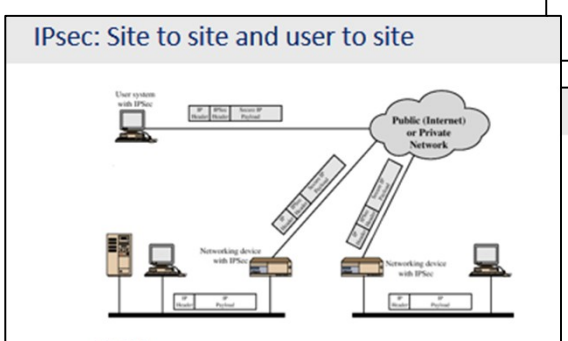
Topics covered

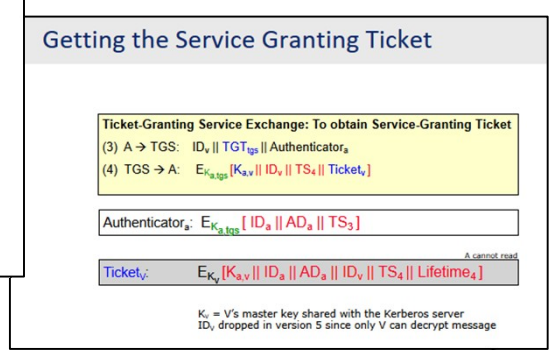
- WLAN security
  - WEP and all its weaknesses
  - WPA, WPA2, WPA3
  - 802.1x - port-based (network) access control
- Security protocols
  - Secure Shell (SSH)
  - Kerberos
  - IPsec: Transport and tunnel mode, ESP and IKE





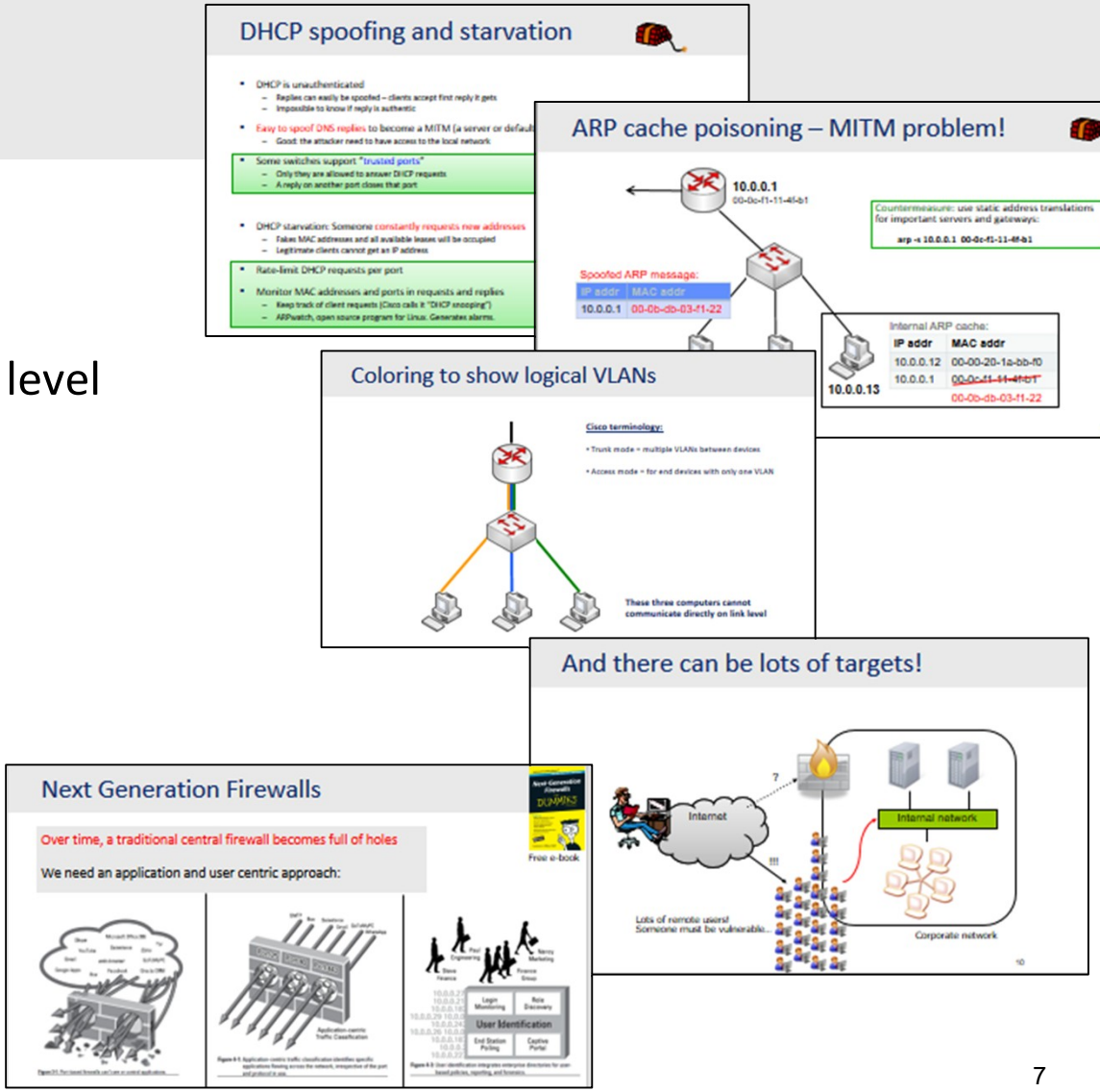






Topics covered

- Link level security
  - Problems and solutions on link level
  - VLAN and switches
- Network design
  - DMZ
  - VPN and remote access
  - Deperimeterization
  - Zero Trust



Some of these tools were discussed in this course – they are useful!

SECTOOLS.ORG

Nmap Security Scanner
- Intro
- Ref Guide
- Install Guide
- Download
- Changelog
- Book
- Docs

Security Lists
- Nmap Announce
- Nmap Dev
- Bugtraq
- Full Disclosure
- Pen Test
- Basics
- More

Security Tools
- Password audit
- Sniffers
- Vuln scanners
- Web scanners
- Wireless
- Exploitation
- Packet crafters
- More

Site News
Advertising
About/Contact

Site Search

Sponsors:

SCAN FOR

Home About/Help Suggest a new tool Search

SecTools.Org: Top 125 Network Security Tools

For more than a decade, the Nmap Project has been cataloguing the network security community's favorite tools. In 2011 this site became much more dynamic, offering ratings, reviews, searching, sorting, and a new tool suggestion form. This site allows open source and commercial tools on any platform, except those tools that we maintain (such as the Nmap Security Scanner, Ncat network connector, and Nping packet manipulator).

We're very impressed by the collective smarts of the security community and we highly recommend reading the whole list and investigating any tools you are unfamiliar with. Click any tool name for more details on that particular application, including the chance to read (and write) reviews. Many site elements are explained by tool tips if you hover your mouse over them. Enjoy!

Tools 1–25 of 125 next page → Sort by: popularity rating release date

Wireshark (#1, ↑1) ★★★★½ (23)

Wireshark (known as Ethereal until a trademark dispute in Summer 2006) is a fantastic open source multi-platform network protocol analyzer. It allows you to examine data from a live network or from a capture file on disk. You can interactively browse the capture data, delving down into just the level of packet detail you need. Wireshark has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session. It also supports hundreds of protocols and media types. A tcpdump-like console version named tshark is included. One word of caution is that Wireshark has suffered from dozens of remotely exploitable security holes, so stay up-to-date and be wary of running it on untrusted or hostile networks (such as security conferences). Read 38 reviews.

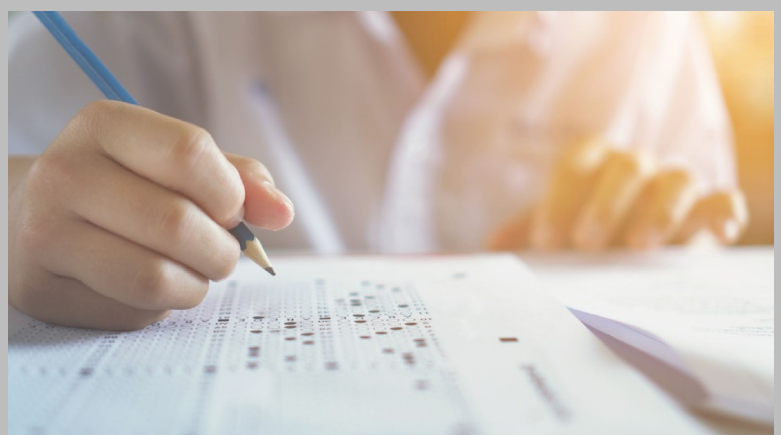
Latest release: version 1.12.7 on Aug. 12, 2015 (2 years, 5 months ago).

sniffers

Metasploit (#2, ↑3) ★★★★½ (10)

Metasploit took the security world by storm when it was released in 2004. It is an advanced open-source platform for developing, testing, and using exploit code. The extensible model through which payloads, encoders, no-op generators, and exploits can be integrated has made it possible to use the Metasploit Framework as an outlet for cutting-edge exploitation research. It ships with hundreds of exploits, as you can see in their list of modules. This makes writing your own exploits easier, and it certainly beats scouring the darkest corners of the Internet for illicit

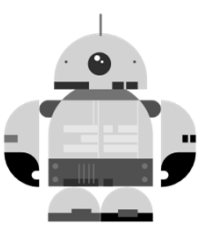
SOME WORDS ABOUT THE EXAM



We will have a digital exam – reexams still traditional

- Bring your own computer
  - Chalmers system uses the Inspera Assessment cloud service for digital-exams
  - Install the Inspera software NOW to make sure it works on your computer
  - If it does not work and you need to book a computer in the examination hall, send an email NOW to the examination administration (see instructions in Canvas)

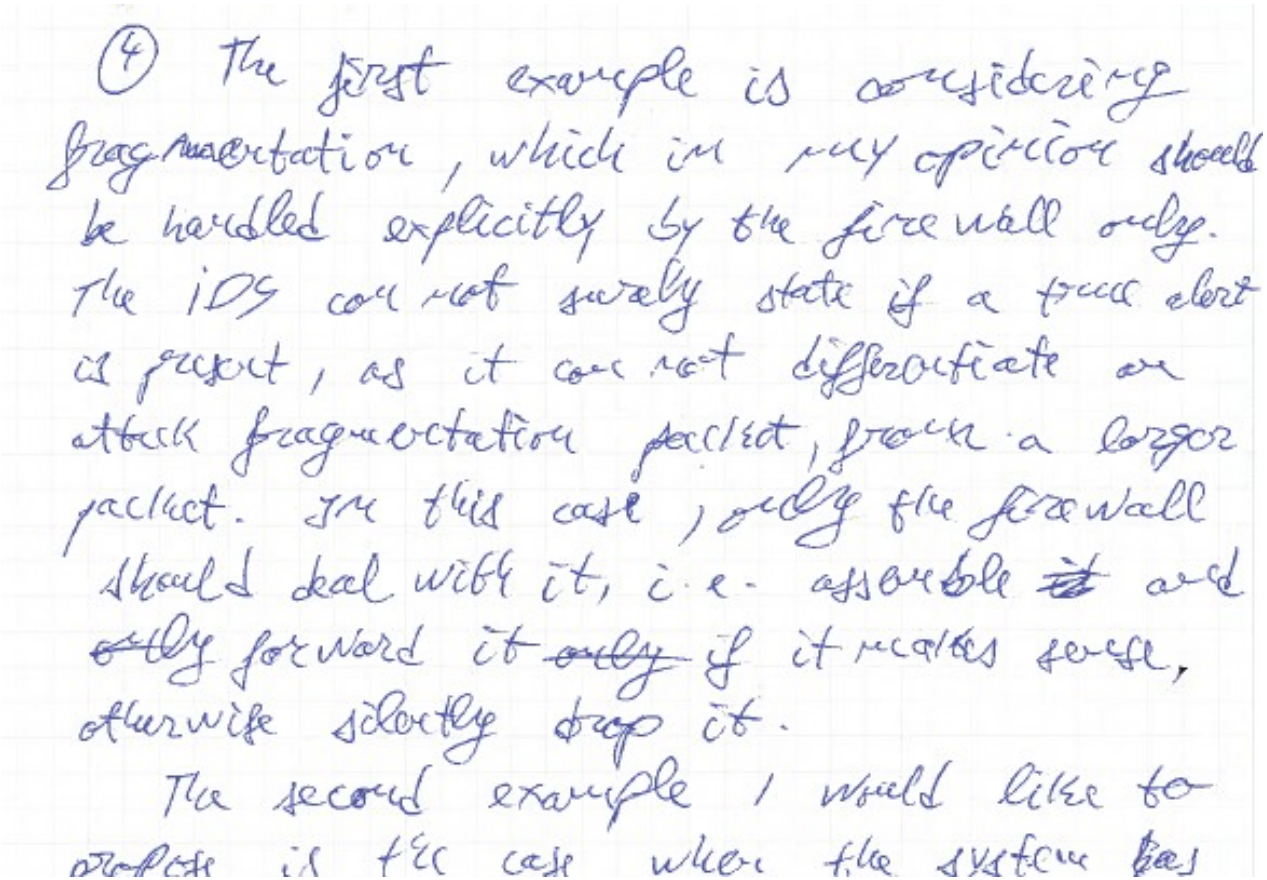
- The exam will be supervised
  - Delete software after exam – may report activities to Inspera
- Arrive in time - preferably 30 minutes in advance to allow for ID checks
- 2 or 3 questions will be automatically corrected when handed in
  - In most of these questions, you get alternatives instead of having to write a motivating answer
  - If close to a grade limit, these questions will be manually inspected
  - There will be space to write additional comments at the end of the exam – if needed



- The rest of the questions will be normal hand-corrected essay questions

https://www.chalmers.se/en/education/your-studies/plan-and-conduct-your-studies/examinations-and-other-summative-assessments/during-and-after-the-examination-session/#digital-examinations-on-chalmers'-campuses

For regular exams – please write readable!



④ The first example is considering fragmentation, which in my opinion should be handled explicitly by the firewall only. The IDS can not surely state if a true alert is present, as it can not differentiate an attack fragmentation packet, from a larger packet. In this case, only the firewall should deal with it, i.e. assemble ~~it~~ and ~~only~~ forward it ~~only~~ if it makes sense, otherwise silently drop it.

The second example I would like to propose is the case when the system has

Some general hints

- No aids are allowed
  - Note that older exams from the pandemic (2021) allowed aids
  - Grades: 3/G: 30p | 4: 39p | 5/VG: 48 (out of 60p)

- General rules:
  - Explain your thoughts – we can not and will not guess!
  - Look at # of points – it tells how much to write
- It is the idea and purpose with a method, algorithm, protocol or parameter that is important – often not details like size, order, etc.
  - Don't drown in the details when you study – try to understand the concepts first
  - When you study a topic, think about why a functionality is present and what the most important purpose(s) are
  - You don't have to memorize headers, etc. If needed, they are given to you

How much should I write?

Q: What is a possible advantage of doing an ACK scan when compared to a SYN scan?

Answers:

1. To see if a host is protected by a stateless firewall. SYN scans will be dropped by a stateul firewall.

2. To see if a host is protected by a stateless firewall. SYN scans will be dropped by a stateful firewall. If the port is closed or ACK is incorrect, a RST is sent as a reply.

3. To see if a host is protected by a stateless firewall. A SYN will be dropped by all types of firewalls, but since a stateless firewall does not keep track of connections, it cannot know whether the ACK belongs to an established session or not and it has to forward it to the host.

   If the port is closed or ACK is incorrect, a RST is sent as a reply. No reply means that the firewall is either stateful or that outgoing RSTs are dropped (we don't know).

But why is the ACK allowed to pass? Is this an answer from someone who thinks the question was obvious or by someone who does not know?

Long motivating answer, the person obviously knows the topic!

Examples from old exams

1. Attacks and DoS

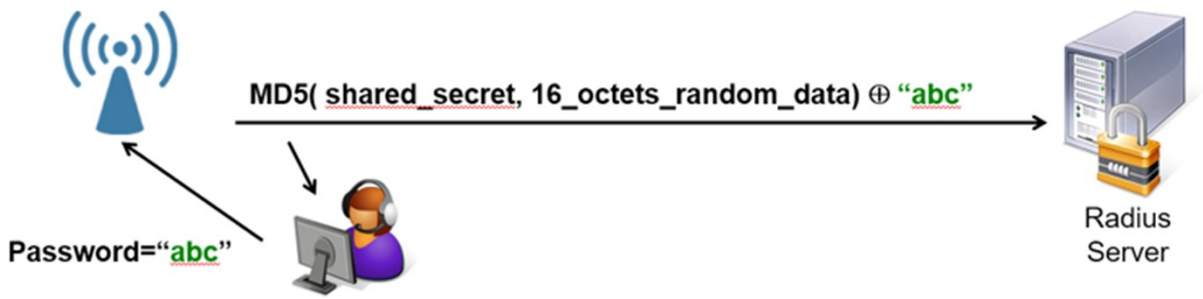
a) There are several ways to scan networks and systems, for example with SYN and ACK scans. Explain the difference(s) between these two types and describe what the expected results from them are. It should be obvious from your answer that there is a reason why scanning tools offer different methods to scan systems. (2p)

SYN scan: shows directly what ports are open and accepts communication (SYN/ACK received) and what ports are closed (RST or no answer).
ACK scan: may work through stateless firewalls: RST means someone (a host) is there, no reply that the port is either filtered by a stateful firewall or there is not host at all.

b) Ping of death is a well-known attack. It sends IP datagrams with a size > 64 kByte. However, the maximum size of an IP datagram is 64kByte due to its 16-bit length field. Explain how is it still possible send such oversized datagrams! (2p)

A naïve implementation would assume IP datagrams never exceed 65,535 bytes since the length field is 16 bits long.
An oversized IP datagram can be created that exceeds this size by sending a fragment with an offset and a length extending the datagram beyond this limit, for example by setting offset to 65,000 and length to 1,000 bytes.

2. Authentication, WLAN



a) In the picture above, we can see an attack against a Radius server. What is the purpose of this attack? What is it the attacker tries to do? Explain! (2p)

The only unknown in the request to the Radius server is the "shared secret" and it may be possible to do (an exhaustive) offline search to find the secret. The secret could be a word from a dictionary, for example.

b) Can a rainbow table be of help to the attacker in the example above? Why/why not? (2p)

It is not useful since the hash (MD5) contains random data. This data is known by the attacker but it is impossible to create pre-calculated tables since each message is unique.

c) WPA3 is the latest WLAN security protocol and it guarantees that all session keys are unique for the session and that they are not in any way related to other session keys. However, the current and most used protocol today, WPA2, does not have this property.
What is this property called? Why is this a good property?
Also explain how this is normally achieved and what "mistake" WPA2 does! (3p)

The property is called (perfect) forward secrecy and means that older and future sessions should not be compromised if a session keys ever broken.
This is normally done by using Diffie-Hellman key agreement to make session keys unique and never reused.
WPA2 calculates session keys from the master key which is derived from the SSID + Password/shared secret (and by an attacker some known random values). If the password is revealed, session keys for all other sessions can be recalculated.

d) Kerberos is a good protocol performing both authentication and authorization. It has some advantages and at least one disadvantage:
1: it uses symmetric keys;
2: the tickets handed over to the clients contains its internal state;
3: however, it requires applications to be Kerberized.

Explain these concepts and what the advantages and disadvantage mean! (3p)

3. Secure Protocols

a) IPsec can operate in both Tunnel and Transport mode. In Tunnel mode, a new IP header has to be inserted. Why?
Also, the new address is not protected by the MAC. Why is this not needed? (2p)

A new header is needed since it is not at the datagram's final destination it is decrypted. The address to the system decrypting the datagram can be modified by someone (MITM) and it will be sent to another destination. However, it is encrypted and useless for the receiver.

b) How is freshness of data packets guaranteed by the TLS record layer protocol? There are no sequence numbers in the header. (2p)

It uses implicit sequence numbers, i.e. both sides counts number of packets and this counter is included when calculating the MAC. This ensures that replays are not possible and that packets are not replays from the past.

c) SSH uses a (keyed) MAC which is calculated from the plaintext data to protect against modification. What is the drawback/danger of using with this method? (2p)

It means that the receiver must decrypt the packet before the MAC can be checked. This is the "Encrypt-and-MAC" method which may cause additional work for a receiver if garbage packets are sent to it. It can also be dangerous (may trigger bugs) if the receiver has to process data packets before it knows they are authentic and correct.

5. TLS and SSH

a) The first TLS messages exchanged between a client and a server are the HELLO messages:

$\{ ver_C \| r_1 \| sid \| ciphers \| comps \}$ → server

$\{ ver \| r_2 \| sid \| cipher \| comp \}$ ← server

Explain clearly what each field contains and the purpose of it (what it is used for)! (4p)

ver_c is the highest TLS version the client supports, ver is what the server decides to use.
r_1 and r_2 are nonces (random numbers) later used in key generation.
sid is session ID, a number identifying the session (a connection can have several sessions). A new session has number 0 from the client.
Ciphers is a list of algorithms the client supports (for encryption, MAC and authentication), in priority order. Cipher is the cipher the server decides to use.
Comps is a list of compression algorithm the client supports, comp is what the server decides to use.

b) A Pseudo-random function is used in TLS:

$$\mathbf{PRF(k, label, x) = HMAC_k(HMAC_k(label||x) \| label||x) \| HMAC_k(\ldots}$$

What is the purpose of this PRF-function? When is it used? (2p)

It is used to expand short secrets to longer blocks, for example to generate the master key from the pre_master_secret or to generate crypto-keys from the master secret:
master_secret = **PRF**(*pre_master_secret*, "**master secret**", $r_c \| r_s$)

Suitable for automatic correction

5. DDoS mitigation

There have been different solutions proposed for how DDoS attacks can be stopped. Suggestions include “traceback”, “pushback” and “centertrack”. Pair the explanations with the corresponding technology! (2p)

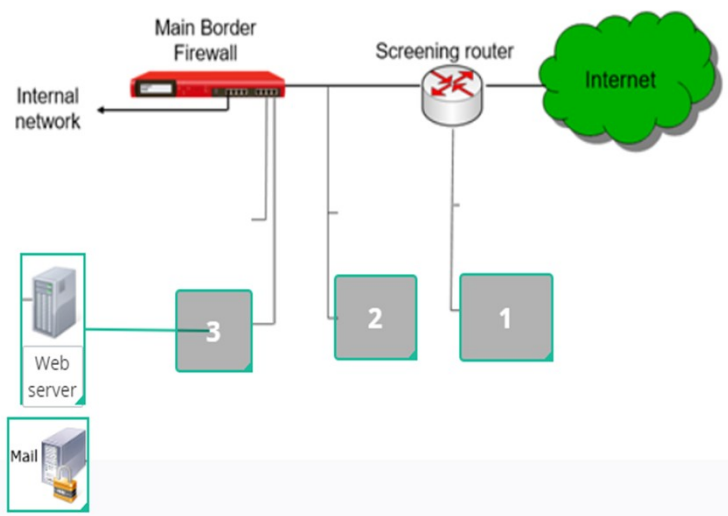
Please match the values:

	Pushback	Centertrack	Traceback
For each datagram forwarded by a router, the receiver may, with some small probability, also receive additional info about who forwarded it,	○	○	○ ✔
Routing within a system (e.g. AS) may be temporarily handled by special routers that are configured to handle tracing, if an attack is seen	○	○ ✔	○
Routers realizing they have to discard many datagrams to one destination can tell other routers to drop these datagrams as well.	○ ✔	○	○

Suitable for automatic correction

Network Design

What is the best placement of a mail and a web server?



What alternative(s) are true about the placement of a web and a mail server?

- [X] Alternative 3 is the most secure placement of a public mail and web server
- [X] Alternative 1 with a DMZ is a very common solution for such servers.
- [X] Alternative 2 should never be used
- [ ] A cracked web server is not critical since the mail server has a built-in firewall.

A correct answer gives +1p, an incorrect -1p.

6. Firewall Technologies

Suitable for automatic correction

Assume we want to block all connections from the Internet to a web server, except for encrypted TLS traffic. The server also has some other services running but they should be hidden and not accessible from the Internet.

Assume the price for the firewall increases for each technology in the order listed 1-5. The cheapest being the static packet filter and the most expensive the air-gap firewall. Choose the cheapest firewall in the list that will fulfill the requirements below!

0,5p for each correct answer, no points for incorrect answers (4p)

1. static packet filters
2. dynamic packet filters
3. deep-packet packet inspection firewall
4. circuit-level gateway
5. air-gap firewalls

NOTE that the order of firewalls is randomized below - they are not in price order!

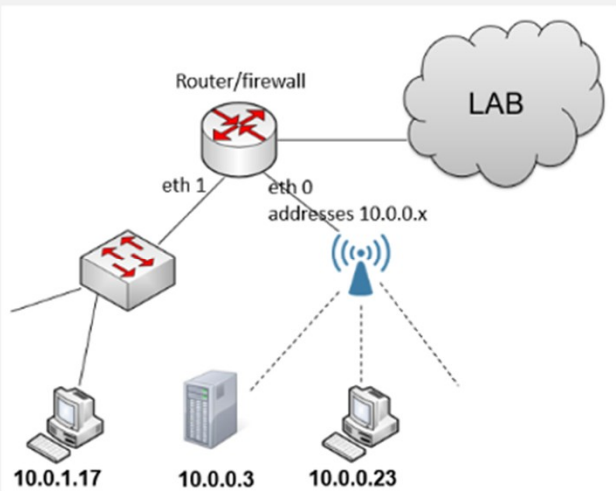
	2. Dynamic packet filter	5. air-gap firewall	4. Circuit-level gateway	None of the firewalls	3. Deep-packet inspection	1. Static packet filter
Replayed TCP segments should not reach the server	○	○	○	○	○ ✔	○
Protect server against SYN flood attacks coming from one and the same IP-address	○ ✔	○	○	○	○	○
DNS requests from the web server and only corresponding replies should be allowed	○	○	○	○	○ ✔	○
SYN-scans should not reveal the hidden services	○	○	○	○	○	○ ✔
Inspect web page requests sent to the server	○	○	○	○ ✔	○	○
ACK or FIN scans should not reveal the hidden services	○	○	○	○	○ ✔	○
Hides the real IP address of the server and protects the server against scanning attacks	○	○	○ ✔	○	○	○
Invalid IP packets with expiring TTLs should not be forwarded	○	○	○	○	○	○ ✔

7. Firewall rules

Suitable for automatic correction

We have a network using the full 10.x.x.x address space, and we want to give LAB access only for the hosts connected to interface eth0 except for the server 10.0.0.3.

Place the firewall rules in the correct order to achieve this! (2p for a correct solution)



iptables -N ip_filter

iptables -A ip_filter -s 10.0.0.3/32 -j DROP

iptables -A ip_filter -i eth0 -s 10.0.0.0/24 -j RETURN

iptables -A ip_filter -s 10.0.0.0/8 -j DROP

How to think when you study

Q: WEP is considered to be insecure. Why?

Step 1: A high level of understanding

What problems have we seen?

- **Keys can be cracked in less than an hour, many attacks are trivial**
- **Crack tools available**
- **Rainbow tables, …**

How does this affect us?

- **We can't use WEP for any serious work**

Step 2 if you have more time:
A more scientific approach

What problems have we seen?

- **IV's can be reused which means …**
- **Uses linear checksum (CRC), this is a problem because …**
- **Caffe Latte attack w. details, …**

Is this a generic problem? What can we learn?

- **Kerberos v4 also used linear checksums**
- **…**

Course evaluation

- Goal is to improve course: contents, lectures, labs, etc.
- **Please spend 5 minutes and make your voice heard!**
- All comments are welcome. Give your impression of the course
  - What is good? What can be improved for next year?
  - How about the labs?
  - Are there new topics you would like to see? Or topics to drop?
  - How is this course compared to other courses? Please tell us where to spend our efforts in improving courses and Master programs

IEEE Spectrum

Vint Cerf on 3 Mistakes He Made in TCP/IP › The co-creator of the Internet's protocols admits his crystal ball had a few cracks

BY TEKLA S. PERRY | 07 MAY 2023 | 2 MIN READ |

https://spectrum.ieee.org/vint-cerf-mistakes

1) “I thought 32 bits ought to be enough for Internet addresses.”

“And of course,” he says, “everybody laughs and says, ‘You idiot, why didn’t you use 128-bit addresses?’ The answer is that, back in 1973, people would’ve said, ‘You’re crazy if you think you need 3.4 times 10 to the 38th addresses to do an experiment that you aren’t sure is going to work.’ So that was a mistake, although I don’t think at the time that I would have been able to sell 128.”

2) “I didn’t pay enough attention to security.”

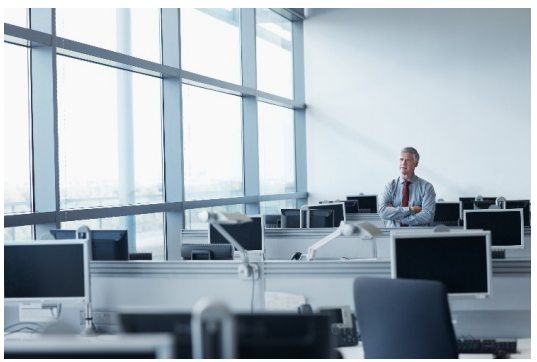
“Before public-key cryptography came around, key distribution was a really messy manual process,” Cerf says. “It was awful, and it didn’t scale. So that’s why I didn’t try to push that into the Internet. And by the time they did implement the RSA algorithm, I was well on my way to freezing the protocol, so I didn’t push the crypto stuff. I still don’t regret that, because graduate students, who were largely the people building and using the Internet, would be the last cohort of people I would rely on to maintain key discipline, though there are times when I wish we had put more end-to-end security in the system to begin with.”

3) “I didn’t really appreciate the implications of the World Wide Web.”

“That is,” Cerf says, “I didn’t expect the avalanche of content that went onto the Internet once the Web was made available. And what happened as a result of that avalanche is that we had to invent search engines in order to find stuff, because there was so much of it. I absolutely did not predict that search engines would be needed.”

Interested in being a Teaching Assistant?

- Why not be a TA in the course next year?
- Sign up as a TA (Amanuens) when you see the advertisements
- If you want to be reminded when ads come out, send an email to <sanna.staf@cse.gu.se> and let her know that you are interested
- It’s fun and you will learn a lot as well ☺



I hope we have reached the goals with the course

- Network security should no longer be a blank paper
- You now have useful tools and enough knowledge to evaluate and design solutions
- Should be useful for the software developer, the designer, the network specialist, the security consultant. For you!

Good Luck!

If you have any questions, just ask! Use email and not Canvas for faster replies.