Exam Advanced Network Security

Jaap-Henk Hoepman, Harald Vranken

June 17, 2019

Name: _____

Student number: S _____

Please answer your questions using the space provided on the exam sheet. Write legibly, and use proper sentences; an unreadable answer is a wrong answer. . . Answer questions concisely, but with sufficient detail and precision. Always explain your answers. Please leave space in the margin for correction marks. Use a separate piece of scratch paper for draft answers, private computations or remarks.

Write your name and student number on every page.

You are not allowed to use books, notes, tablets, PC's and (smart)phones during the exam.

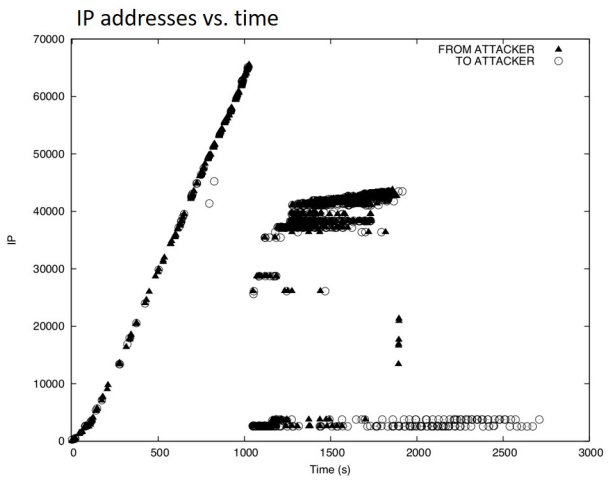
The total number of points that can be scored is 60, as specified alongside the questions. The final grade equals $1 + 9 * \frac{points}{60}$, rounded to the nearest half grade (except for grades between 5 and 6 which are rounded to nearest full grade).

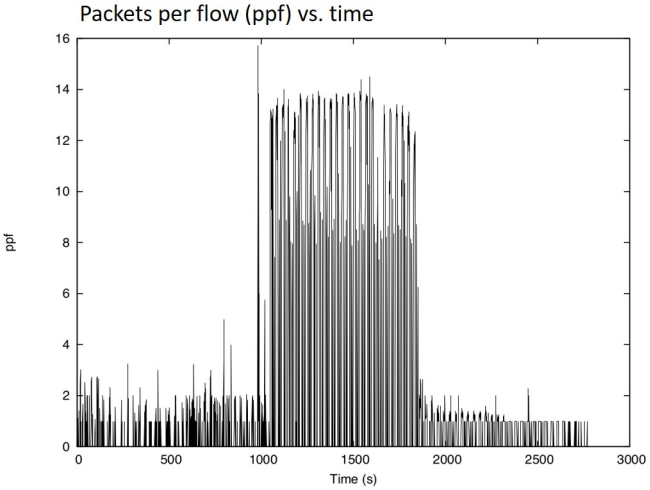
Good luck!

Question 1 (5 points): Preventing and detecting network attacks

a. *(2 points)* The first stage of a typical flow monitoring archictecure is packet observation, in which network packets are captured and timestamped. In addition, packets may be pre-processed before they are passed to the subsequent flow metering & export stage. What pre-processing steps can be taken, and for what reason?

b. *(3 points)* The figures below show flow data for a certain type of attack. In the figure on the left, each mark represents a connection from the attacker to a victim or vice versa. In the figure on the right, each mark represents the number packets per flow. What type of attack are we facing, and what attack phases can be distinguished?





Answer:

a. The pre-processing steps are packet truncation, sampling, and filtering. Truncation selects only those bytes of a packet that fit into a pre-configured snapshot length. Sampling and filtering select a subset of the packets. Sampling (either systematic or random) selects some of the packets, while still being able to estimate properties of the full packet stream. Filtering removes all packets that are not of interest. These steps reduce the amount of data to be received and processed in subsequent stages.

b. This is a SSH dictionary attack. We can identify three attack phases. The first phase is the scanning phase (first 1000 seconds) in which the attacker performs a sequential SSH scan spanning over the entire IP4 address space to gather information on which hosts run a vulnerable SSH service. Only few victims respond to the attack. The scanning phase is characterized by only few packets per flow (typically only a three-way handshake for the responding victims). The second phase is the attack phase (next 1000 seconds) in which the attacker initiates a brute-force user/password guessing attack. In this phase, only a small subset of the hosts in the network is involved, and the number of packets per flow has a sharp rise. The third phase (after 2000 seconds) contains residual traffic in which the attacker communicates with the compromised hosts. We now see only few packets per flow.

Question 2 (5 points): Economics of network security

a. *(3 points)* Why can online banking services that become unavailable due to DDoS attacks be considered as a tragedy of the commons? How could this be solved?

b. *(2 points)* What does the statement 'CVSS is DoS-ing your patching' mean?

Answer:

a. Individual users are willing to pay for security measures that protect their own systems, but they are unlikely to spend money on security measures to protect other systems (in this case systems operating online banking services). When the user systems are exploited in DDoS attacks against banks, the user systems themselves are not harmed. However such DDoS attacks cause banking services to become unavailable to all users. The solution is regulatory rather than technical. The costs of DDoS attacks could for instance fall on the operators of the networks from which the attacks originate, which is an incentive to put pressure on their customers to install suitable defensive software, or to supply it themselves.

b. The number of vulnerabilities reported is huge. There are incentives, both for security researchers and security suppliers, to assign high CVSS score to reported vulnerabilities. Hence, users are requested to install large numbers of patches to repair critical vulnerabilities on a daily basis. In practice, this is unfeasible and hence users will not do so. Furthermore, many high-rated vulnerabilities are never actually exploited, which further decreases the willingness of users to patch.

Question 3 (5 points): WiFi security

a. *(2 points)* Two main steps in WPA2 Enterprise are 802.1x authentication followed by a 4-way handshake. What is the purpose of these steps?

b. *(3 points)* The 4-way handshake is vulnerable to a key reinstallation attack. How does this attack work?

Answer:

a. First, 802.1x provides mutual authentication between a device (user) and an access point (AP). At the end of this step, both parties know the PMK. Next, in the 4-way handshake the device and AP agree on a session key (PTK).

b. The attack exploits a flaw in the 4-way handshake, which allows the device to accept retransmissions of message 3, which forces a reinstallation of the PTK. First a man-in-the-middle (MitM) is positioned between the device and AP. The MitM prevents message 4 from arriving at the AP, which triggers retransmissions of message 3 by the AP. On reception of the retransmitted message 3, the device will reinstall the PTK. The actual flaw is that this reinstallation also resets associated parameters, such as the nonce used in the data-confidentiality protocol and the replay counter. Depending on which protocol is used, this allows an attacker to replay, decrypt, and/or forge packets.

Question 4 (5 points): Mobile telephony security

a. *(3 points)* At least the very first time a SIM in a GSM identifies itself to the network, the SIM will transmit its unique identifier, the IMSI (International Mobile Subscriber Identity). When this identifier is transmitted in plain-text over the wireless network, it can easily be intercepted (IMSI catching). Can IMSI catching be prevented by encrypting the IMSI with the public key of the home network?

b. *(2 points)* When eavesdropping on GSM, the first step required is to capture the GSM signals. To what extend does channel hopping prevent eavesdropping?

Answer:

a. This has a practical limitation, since the encrypted IMSI would not fit inside the currently defined identity response messages. Furthermore, the encrypted IMSI can simply be intercepted as well by an attacker and re-used. It is unfeasible to change the public key for every identification. This can be resolved by adding some randomness to every encryption of the IMSI. This is actually done when replacing the IMSI with a Pseudo Mobile Subscriber Identifier (PMSI).

b. Channel hopping is used as a signal quality measure and causes the transmission to switch to a new frequency after every single burst. The challenge in capturing the GSM signals lies in receiving the bursts on time and in demodulating them correctly (ie., following the channel hopping sequence with enough precision). While not a security measure on itelf, the hopping sequence, which is often negotiated confidentially, makes eavesdropping on GSM frequencies much harder and requires breaking the encryption really fast in order to follow the hopping sequence in time.

Question 5 (5 points): Routing security

a. *(3 points)* BGPsec provides origin authentication, which ensures that an AS only announce prefixes that are assigned to it. How is origin authentication provided?

b. *(2 points)* SCION uses two levels of routing, intra-ISD and inter-ISD. Both levels utilize path-segment construction beacons (PCBs) to explore routing paths. How are PCBs secured?

Answer:

a. Resource PKI (RPKI) provides origin authentication using certificates. When receiving a prefix announcement, the receiver of this announcement will look up the ROA (Route Origin Authorization) of this prefix in the RPKI repository. The ROA is a signed object by which the address space holder authorizes the AS to announce prefixes for (a set of) that address space. Hence, the prefix and AS in the announcement should match with the ROA. The signature of the ROA can be verified by the corresponding End-Entity certificate (which attests the IP addresses that have been allocated to the address space holder). The signature of the EE certificate can be verified by the CA certificate, and the IP addresses in the EE certificate should be a subset of the IP addresses in the CA certificate. This is repeated for all CA certificates in the chain up to the root CA certificate (trust anchor).

b. A core AS announces a PCB and disseminates it within an ISD or among core ASes (beaconing). Each AS signs the PCB it forwards. This signature enables PCB validation by all entities. Certificate servers store cached copies of ASes' certificates. Certificate servers are queried by beacon servers when validating the authenticity of PCB.

Question 6 (5 points): Botnets

a. *(3 points)* A botnet can apply domain fluxing by means of DGA (Domain name Generation Algorithm). Why does this evade detection of the botnet?

b. *(2 points)* Why do botnets like to use Dynamic DNS?

Answer:

a. The evasion strategy of the botnet is to frequently use a new domain name for the C&C-server. In this strategy, DGA helps to evade what this new domain names will be. Bots periodically generate a (large) number of pseudo-random domain names, using some domain name generation algorithm. Not all generated domain names are active at a given time, and only few are actually registered by the botmaster and correspond to C&C-servers. When re-engineering the bot malware, we can also re-engineer the DGA and hence predict what domain names will be used by the botnet in future. However, it is nearly unfeasible to register all those domains by law enforcement (to take over the botnet), or to block them (to stop the botnet), or even to check which ones are malicious.

b. DDNS performs domain name-to-IP mapping with dynamic IP addresses. DDNS services provide automatic reconfiguration of DNS, typically when the IP address handed out by an ISP is changed. DDNS facilitates IP flux, meaning that IP addresses are changed fast. Botnets make use of DDNS to keep the C&C server domain name to IP address mapping up to date in real-time.

Question 7 (15 points):

a. *(6 points)* What are the three properties a mutual exclusion protocol must satisfy? Give their name and their definition.

b. *(9 points)* Consider a unidirectional ring of n processors that communicate by message passing. I.e. node i can send messages to node $i+1 \pmod n$. Write a mutual exclusion protocol for this system. (Note: you may assume all nodes, all links and all memory always work correctly; no need to consider fault tolerance). Prove your protocol correct.

Answer:

a. A mutual exclusion protocol needs to satisfy the following three properties: *Mutual exclusion*: there is at most one processor in the critical section, *Progress*: if there is at least one processor enters, and the critical section is empty, then one of these processors will eventually get access to the critical section, and *No starvation*: if a processor enters, and if all processors that get access to the critical section release it, then it will eventually get access. (Answers must be precise! Two point per property.)

b. A mutual exclusion protocol generally looks like this:

initialisation
while *true*
do *enter*
critical section
release
remainder section

If we do not have to consider faults, the following simple protocol will do.

Node 0 has the token
while *true*
do while not having the token
do wait to receive the token
critical section
send the token to the clockwise neighbour
while in remainder section
do if your receive a token from the counterclockwise neighbour,
forward the token to the clockwise neighbour

To prove correctness, observe the following.

Mutual exclusion: Nodes only enter the critical section if they have the token. As there is exactly one token in circulation, there is at most one processor in the critical section.

Progress: If there is at least one processor entering, and the critical section is empty, then the token is held by a node that is either entering or in the remainder section. If the node is entering, this node will eventually get access to the critical section. If the node is in the remainder section, it will forward the token to the clockwise neighbour, thus moving the token one step closer to a processor entering. Eventually the token will reach one of these processors, allowing it to enter.

No starvation: if a processor enters and doesn't have the token yet, and if all processors that get access to the critical section release it, then the token will be moved in clockwise direction along the ring until it reaches this particular processor.

(Three points for the protocol; two points per proof obligation. Points deducted for extra assumptions (like FIFO, or time). Point deducted if not properly initialised.)

Question 8 (15 points): Consider a graph with two nodes 0 and 1 and an edge e between them. Let $C[i]$ be the state of node i, that can be read by node $1-i$. Let each node store a value $C[i].dir \in \{0,1,2\}$. We define e to be directed from i to $1-i$ if and only if

$$C[i].dir + 1 = C[1-i].dir \pmod 3 .$$

The following self-stabilizing protocol allows 0 and 1 to direct the edge e between them under a distributed daemon (the code shown is the code for node i; all three lines are executed as one single statement):

$d \leftarrow C[1-i].dir$
if $C[i].dir = d$
then $C[i].dir \leftarrow d + 2 * i - 1 \pmod 3$.

(where d is a local variable).

a. *(3 points)* What are the legitimate states of this protocol?

b. *(6 points)* Show that this protocol is self-stabilising under the distributed daemon.

c. *(6 points)* Under which condition(s) would the protocol still stabilise if the state space is reduced and $C[i].dir$ takes values only in $\{0,1\}$ (and assignments and comparisons are done modulo 2 instead of 3)? Why?

Answer:

a. The legitimate states are all states where $C[0].dir \neq C[1].dir$.

b. We have to show *closure* and *convergence*. (These must be explicitly named!) Closure is trivial: once $C[0].dir \neq C[1].dir$ none of the two nodes takes another step that changes it's state. Convergence is proven as follows. Let us write (x,y), with $x,y \in \{0,1,2\}$, as a shorthand for the configuration where $C[0].dir = x$ and $C[1].dir = y$. Let $x = y$ (otherwise we are already in a legitimate state).

   If either 0 or 1 takes a step (but not both), the system reaches a legitimate state: if 0 takes a step, then the new state becomes $(x',y') = (-y1 \pmod 3, y)$ while if 1 takes a step, then the new state becomes $(x',y') = (x, x+1 \pmod 3)$. In both cases, if $x = y$ then $x' \neq y'$.

   If both 0 or 1 take a step simultaneously (which is possible under the distributed daemon) the system reaches a legitimate state as well: the new state becomes $(x',y') = (y-1 \pmod 3, x+1 \pmod 3)$. If $x = y$ then $x' \neq y'$.

c. If $C[i].dir$ takes values only in $\{0,1\}$ the protocol no longer converges under the distributed daemon. It does still converge under the central daemon as inspection of the proof of the previous question reveals.

   (Alternative answers: under the assumption that at some point not both processors take a step simultaneously; assuming execution of the code is atomic - as an alternative way to say we have a central daemon.)

(end of exam)