# ECE 498 MP3 Report Template

Spring 2019

<qnazir2, navjot2, shuyuel2>

# Task 0 – identify the malicious traffic

- The attacker pcap file: http.pcap
- Reason:
  - For example:
    - the *Content-Type* in pcap file for legitimate traffic _does not contain a '#cmd' string__, while the *Content-Type* in pcap file for malicious traffic from an attacker __contains a '#cmd' string____.

# Task 1 – HTTP Traffic Analysis: discovery

- The UNIX timestamp of the first attempted scan:
  - 1521394903.610774
- IP address of the vulnerable server:
  - 172.17.0.2
- Port of the vulnerable server:
  - 8080

# Task 1 – HTTP Traffic Analysis

- Number of unique content type headers: 14
- Minimum length  and Maximum Lengths: min: 33 max: 845
- Table from Subtask 2.b

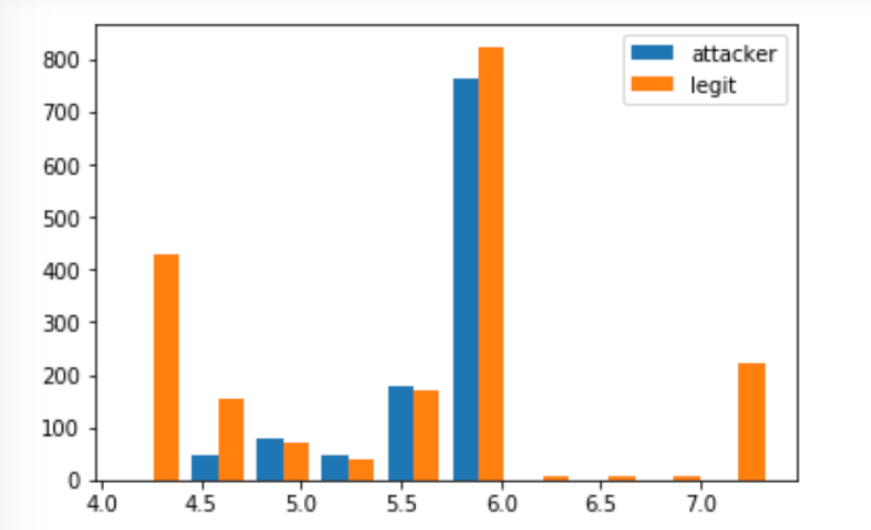
| Command Name | Present in the attack? | Interpretation of the command |
|---|---|---|
| **whoami** | Yes | Displays the name of the current user |
| **wget** | Yes | A free utility for  non-interactive download of files from the web |
| **ls** | Yes | Displays the names of files contained within that directory |
| **cat** | No | Print file or input on the standard output. |
| **cd** | No | Change working directory. |
| **insmod** | Yes | Simple program to insert a module into the Linux Kernel. |
| **ssh** | No | Remotely connect to server. |
| **lsmod** | No | Displays which loadable kernel modules are currently loaded. |

# Task 1 – Host Log Analysis

- Number of unique kernel modules: 303
- Attacker controlled module: 4
- How did you verify the attacker controlled module was loaded? Since it's showed 'added' above, then it's loaded.
- What is the file name that contains the internal hostnames that the attacker use for lateral movement? know_hosts.swp & known_hosts.swpx
- Do you observe any evidence that the attacker extracted the internal host names via HTTP in the logs? There is not a naïve attacker since 'know_hosts' files don't exist in content-type.

# Task 1 – DNS Traffic Analysis

- Attacker controlled DNS server:
  - 162.212.156.148
- Legitimate DNS server:
  - 10.0.2.15
- Histogram of length of queries for both servers:



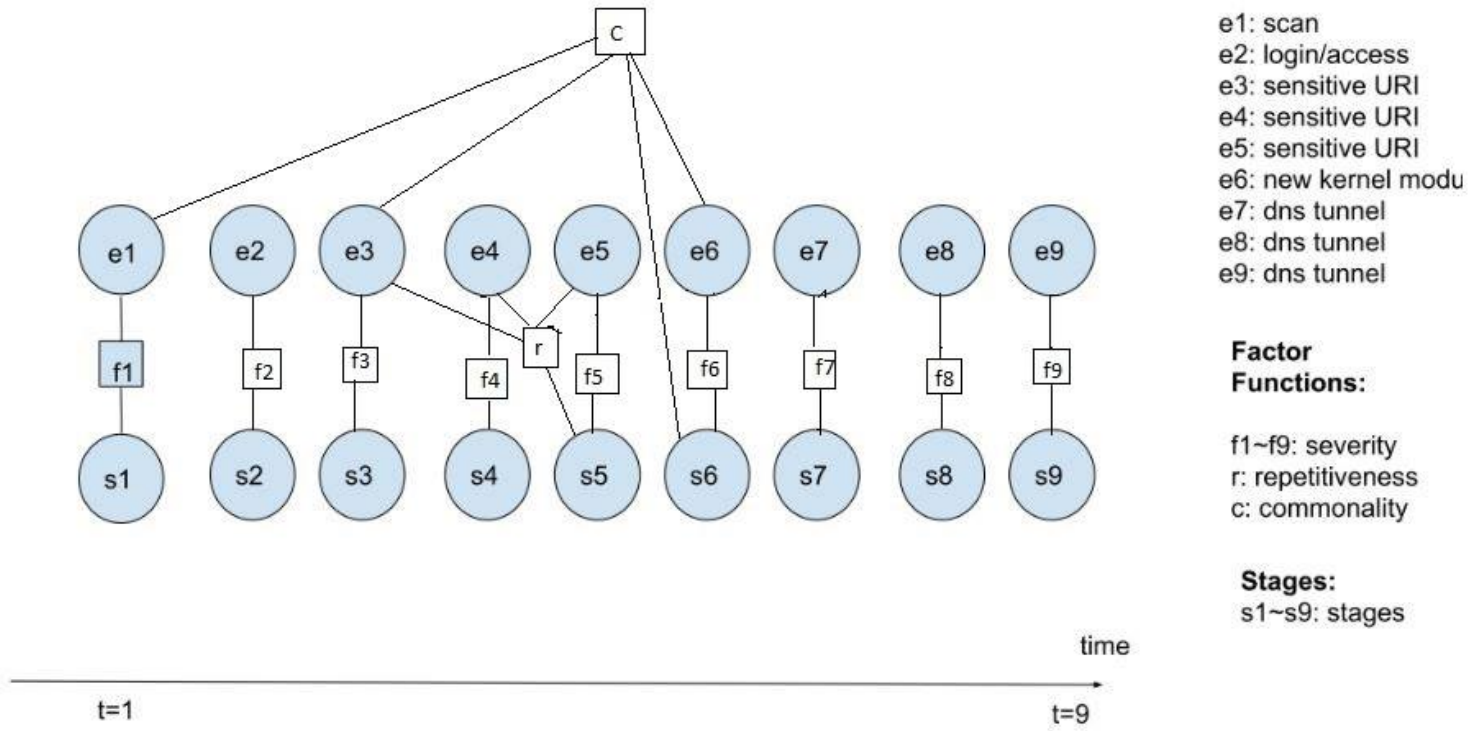
# Task 2 – Simple Factor Graph

- Marginal Probability $P(S_1)$:
  - P{S1=0} = 0.69387755
  - P{S1=1} = 0.30612245
- Value of $S_1$ that maximizes $P(S_1)$:
  - S1 = 0
- Hand calculations:

- step 1:
  - $$P\{S1 = 0, E1 = 0\} = \frac{1}{Z} * 0.85 * 0 = 0$$
  - $$P\{S1 = 0, E1 = 1\} = \frac{1}{Z} * 0.85 * 0.2 = \frac{0.17}{Z}$$
  - $$P\{S1 = 1, E1 = 0\} = \frac{1}{Z} * 0.15 * 0 = 0$$
  - $$P\{S1 = 1, E1 = 1\} = \frac{1}{Z} * 0.15 * 0.5 = \frac{0.075}{Z}$$
- step 2:
  - $$Z = 0 + 0.17 + 0 + 0.075 = 0.245$$
- step 3:
  - $$P\{S1 = 0, E1 = 0\} = \frac{1}{Z} * 0.85 * 0 = 0$$
  - $$P\{S1 = 0, E1 = 1\} = \frac{1}{Z} * 0.85 * 0.2 = \frac{0.17}{Z} = \frac{0.17}{0.245} = 0.69387755$$
  - $$P\{S1 = 1, E1 = 0\} = \frac{1}{Z} * 0.15 * 0 = 0$$
  - $$P\{S1 = 1, E1 = 1\} = \frac{1}{Z} * 0.15 * 0.5 = \frac{0.075}{Z} = \frac{0.075}{0.245} = 0.30612245$$

# Task 3 – Factor Graph Inference (3.0)

- Image of factor graph defined in your code:




**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

**Factor Functions:**

f1~f9: severity
r: repetitiveness
c: commonality

**Stages:**
s1~s9: stages

# Task 3 – Factor Graph Inference (3.1-3.4)

- A table containing
  - Inference on hidden states from $t = 1$ to $t = 9$
  - Recommended action for each state at each timestamp from $t = 1$ to $t = 9$

| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Hidden States | discovery | benign | benign | benign | Privilege escalation | persistence | exfiltration | exfiltration | exfiltration |
| Recommended Action | NO_OP | NO_OP | NO_OP | NO_OP | MONITOR | MONITOR | STOP | STOP | STOP |

- How did you pick the recommended action for each step?
  - We pick the recommended action for each step by picking the action with the maximum probability.

- At what time step should you stop the attack?
  - The earliest stage at which the model recommends the 'STOP' action is at S7.

# Task 3 (3.5)

- **Subtask 3.5** Inference result of the most probable states on the new factor graph that excludes $e_7$ and $s_7$.

| t | 1 | 2 | 3 | 4 | 5 | 6 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| Hidden State | discovery | benign | benign | benign | Privilege escalation | persistence | exfiltration | exfiltration |

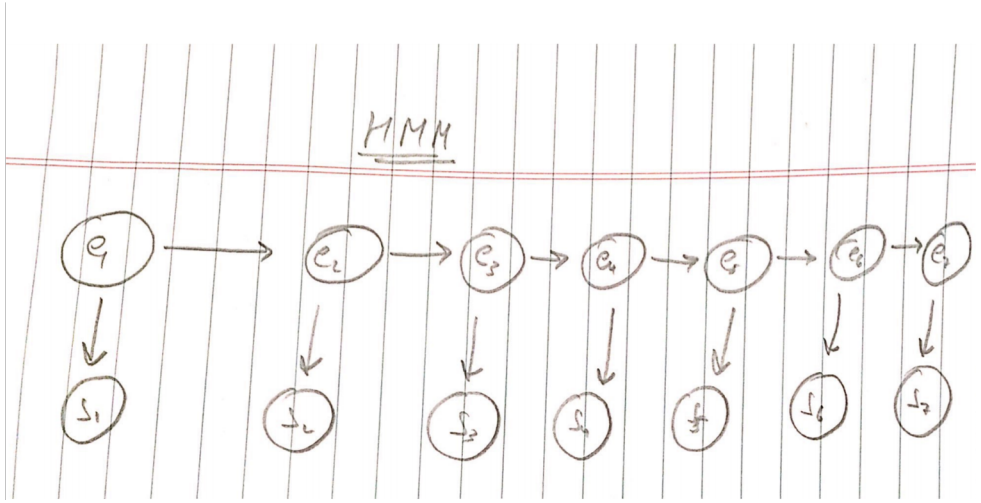
- Compare the most probable states with those in Subtask3.2.

  Are they the same or different? Explain.

  It remains same since there is no connection between E7, S7 with others. In other words, the marginals stay the same as removing e7 and s7 does not affect any other state as there is no factor function connecting it to any other one.

# Task 3 (3.6)

- Visual representation of the HMM model for the provided attack scenario




- Parameters needed for this HMM model to work:
  - Transition state matrix, Emission state matrix, Initial probabilities

# Task 3 (3.6)-Cont.

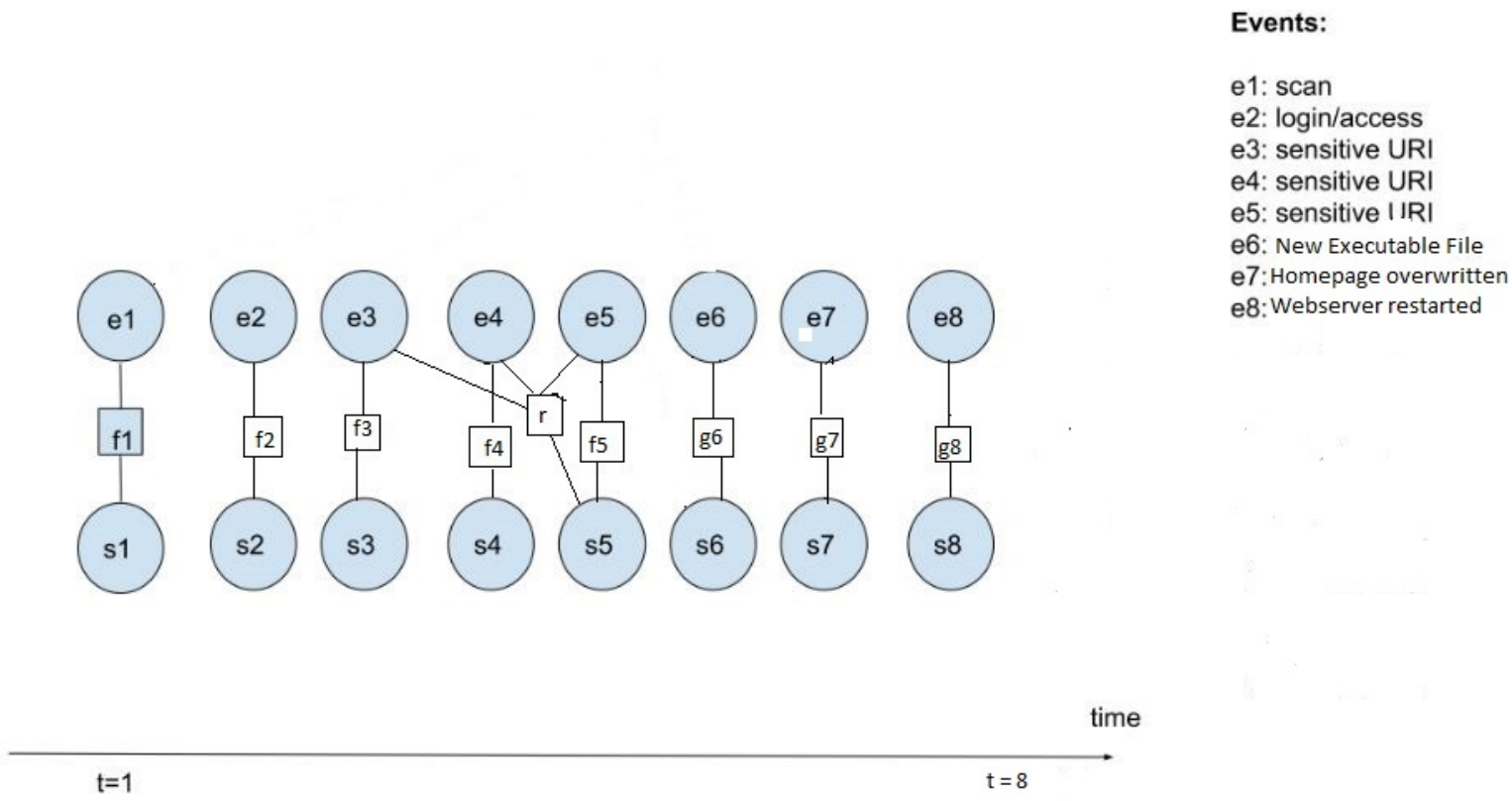
- Give an example of an advantage of the FG over the HMM model.
  - One advantage of Factor graphs over the HMM is that it is more general, i.e., HMM has the markov assumption considering that the future state only depends on the present state. There is no such assumption in Factor graphs and we can have factor functions for modelling any number of dependencies.

# Task 4 (0)

- **Subtask 4.0** Is it possible to *accurately* detect this attack using only one, e.g., $\epsilon_1$, of the five listed events? For each the five listed events, give an example of a situation in which a false positive (i.e., mis-detecting a legitimate user as an attacker) could happen.
  - It is not possible to this attack using only one event as we need the whole sequence of events.
  - Scan: If we get the probability of Discovery higher than Benign but it's a legitimate user
  - Login: Cannot get a false positive as there is no other state
  - Sensitive URI:  If we get the probability of privilege escalation higher than Benign but it's a legitimate user
  - New Executable File: If we get the probability of persistence higher than Benign but it's a legitimate user
  - Homepage overwritten with a new link: If we predict command and control or execution but it's Benign and a legitimate user
  - Webserver restarted: If we predict command and control or execution but it's Benign and a legitimate user

# Task 4 (1)

- **Subtask 4.1** Visual representation of the factor graph.




**Events:**

e1: scan
e2: login/access
e3: sensitive URI
e4: sensitive URI
e5: sensitive URI
e6: New Executable File
e7: Homepage overwritten
e8: Webserver restarted

# Task 4 (2)

- **Subtask 4.2** What variables and factor functions are common to the factor graph in Task 3 and your factor graph in 4.1? *Hint: Are any events common to both factor graphs? Repeating?*
  - The variables Scan, Login and Sensitive URI and the corresponding factor functions (f1, f2, f3, f4, f5 and r) are the same as the previous factor graph.

# Breakdown of Contribution

- Qnazir2 (33%):
- Navjot2 (33%):
- Shuyuel2 (33%):
    - Our team work on MP3 together in library.