Waterford
Institute of
Technology

PARSLEY MANAGEMENT SYSTEM – THREAT MODEL

Secure Programming and Scripting

ABSTRACT

Threat modelling on a new software Parsley Management System, an employee management system for use by small to medium sized companies.

Samantha Sheehan
Computer Forensics and Security

Contents

Introduction 2

System Requirements 2

Employee 2

Features 2

Use Cases 2

Manager 3

Features 3

Use Cases 3

Payroll Technician 4

Features 4

Use Cases 5

System Administrator 6

Features 6

Use Cases 6

Use Case Diagram (created at draw.io) 7

Threats 8

Internal Threat – Disgruntled Employee 9

Abuse Cases 9

External Threat – System Hacker 10

1. Staff negligence 10
2. Brute Force Password Attack 10
3. Dictionary Attack 10
4. Key logger Attack 10

Abuse Cases 10

Misuse Case Diagram (created at draw.io) 11

Threat Mitigation 12

Validation 12

Use Secure Methods/Libraries 12

Educate users 12

Two Signature Authentication for Important 12

Persistent Penetration Testing 12

Introduction

System Requirements

Employee

Features

- Access Features:
    - ➢ Personal information – pps no., employee no. etc.
    - ➢ Upcoming payment – hours worked, hourly rate, payment.
    - ➢ Current hourly rate – if on a wage.
    - ➢ Current tax payments – PRSI, PAYE.
    - ➢ Display/download work roster
    - ➢ Holiday hours used/available
    - ➢ Commission earned – if applicable
    - ➢ Commission rates – if applicable
- Add/Edit Features:
    - ➢ Contact information
    - ➢ Bank account information
    - ➢ Login password
- Specialised Requirements:
    1. None
- Report Issues

Use Cases

Example: Employee logs into system to check his/her roster

1. Employee opens software to homepage
2. Employee is prompted to login
3. Employee enters employee number/username and password
4. Outcomes:
    - ➢ Login attempt successful - Employee is brought to the employee specific interface
    - ➢ Login attempt unsuccessful – Employee is prompted to re-enter login details or contact the system administrator
5. Employee clicks on the ‘Roster’ tab
6. Employee is re-directed to roster folder, where the departments roster is on display

Example: Employee logs into system to change e-mail address

1. Employee opens software to homepage
2. Employee is prompted to login
3. Employee enters employee number/username and password
4. Outcomes:
    - ➢ Login attempt successful - Employee is brought to the employee specific interface
    - ➢ Login attempt unsuccessful – Employee is prompted to re-enter login details or contact the system administrator
5. Employee clicks on the ‘Contact Details’ tab
6. Employee clicks on the ‘Update contact information’ tab
7. Employee clicks on ‘E-mail address’ tab
8. Employee is re-directed to a form page and prompted to enter new e-mail address
9. Employee clicks on submit
10. E-mail address is updated

Manager

Features

- Access Features:
  Personal
  - ➢ Personal information – pps no., employee no. etc.
  - ➢ Upcoming payment – hours worked, hourly rate, payment/salary.
  - ➢ Current hourly rate / Salary
  - ➢ Bonus targets – specification, attained/unattained
  - ➢ Current tax payments – PRSI, PAYE.
  - ➢ Holiday hours used/available

  Staff
  - ➢ Contact information
  - ➢ Hours Worked
  - ➢ Personnel file
  - ➢ Hourly rates/salary
  - ➢ Commission rates – if applicable
  - ➢ Targets – sales/project completion/performance
  - ➢ Performance records – achievements, punctuality, previous performance reviews
- Add/Edit Features:
  Personal
  - ➢ Login password
  - ➢ Contact information
  - ➢ Bank account information

  Staff
  - ➢ Roster
  - ➢ Break schedule
  - ➢ Holiday schedule
  - ➢ Hours worked
  - ➢ Performance – target/goal completion, punctuality, absence etc.
  - ➢ Commission earned – if applicable
  - ➢ Holiday hours – used/available
  - ➢ Illness – days taken
- Specialised Requirements:
  - ➢ Assign a staff account to their department – with approval from the system administrator
  - ➢ Remove a staff account from their department – with approval from the system administrator
  - ➢ Upload files- .xls for roster, holiday schedule etc., .pdf files from employee's performance review, .jpeg or .png files for images of illness certificates, appointments etc.
- Report Issues

Use Cases

Example: Manager logs into system to access an employee's contact information

1. Manager opens software to homepage
2. Manager is prompted to login
3. Manager enters employee number/username and password
4. Outcomes:
   - ➢ Login attempt successful - Manager is brought to the manager specific interface
   - ➢ Login attempt unsuccessful – Manager is prompted to re-enter login details or contact the system administrator

5. Manager clicks ‘Employee Details’ tab
6. Manager selects an employee from a list
7. Manager is redirected
8. Employees information is displayed

Example: Manager logs into system to add the department roster

1. Manager opens software to homepage
2. Manager is prompted to login
3. Manager enters employee number/username and password
4. Outcomes:
    - Login attempt successful - Manager is brought to the manager specific interface
    - Login attempt unsuccessful – Manager is prompted to re-enter login details or contact the system administrator
5. Manager clicks on ‘Roster’ tab which displays options- edit roster, delete roster, add new roster
6. Manager selects ‘add new roster’ from options
7. Manager is prompted to add file path or browse the system
8. Manager selects browse and selects the file on the computer
9. Manager selects upload
10. Outcomes:
    - File uploads successfully – Manager selects save file
    - File upload unsuccessful – Manager is prompted to try again or contact the system administrator
11. Manager is prompted ‘make file available to view?’
12. Manager clicks ‘yes’ or ‘save for later’

Payroll Technician

Features

- Access Features:

  Personal
    - Personal information – pps no., employee no. etc
    - Upcoming payment – hours worked, hourly rate, payment/salary.
    - Current hourly rate / Salary
    - Bonus targets – specification, attained/unattained
    - Current tax payments – PRSI, PAYE.
    - Holiday hours used/available

  Staff
    - Staff rosters
    - Hours worked
    - Holiday hours – used/available
    - Commission earned
    - Bonus – targets/goals achieved
- Add/Edit Features:

  Personal
    - Login password
    - Contact information
    - Bank account information

  Staff
    - Personal information – pps no., employee no. etc.
    - Upcoming payment – hours worked, hourly rate, payment/salary.
    - Commission rates – if applicable
    - Current hourly rate/ Yearly salary

  - ➢ Current tax payments – PRSI, PAYE.
  - ➢ Holiday payment – with approval from the system administrator
  - ➢ Illness payment – with approval from the system administrator
- Specialised Requirements:
  - ➢ Create a new staff account – with approval from the system administrator
  - ➢ Remove a staff account from the system – with approval from the system administrator
  - ➢ Print employee's payment information/tax details
- Report Issues

Use Cases

Example: Payroll technician logs into system to amend an employee's wages to include illness payment

1. Payroll technician opens software to homepage
2. Payroll technician is prompted to login
3. Payroll technician enters employee number/username and password
4. Outcomes:
   - ➢ Login attempt successful - Payroll technician is brought to the payroll technician specific interface
   - ➢ Login attempt unsuccessful – Payroll technician is prompted to re-enter login details or contact the system administrator
5. Payroll technician clicks on 'Department' tab which displays options
6. Payroll technician clicks on 'Employee Details' tab
7. Payroll technician selects specific employee from a list
8. Payroll technician clicks on 'edit payment information'
9. Payroll technician clicks on 'add illness pay'
10. Payroll technician is prompted to enter number of hours' employee has been out sick for
11. Payroll technician enters the correct number of hours and clicks 'submit'
12. The payment is calculated according to the set illness payment and number of hours and displayed on a prompt tab
13. Payroll technician selects save
14. Payroll technician is prompted 'Send approval notification' and clicks 'yes'
15. Prompt screen displays 'Waiting for Approval'
16. Payroll technician selects 'OK'

Example: Payroll technician logs into system to create a new staff account

1. Payroll technician opens software to homepage
2. Payroll technician is prompted to login
3. Payroll technician enters employee number/username and password
4. Outcomes:
   - ➢ Login attempt successful - Payroll technician is brought to the payroll technician specific interface
   - ➢ Login attempt unsuccessful – Payroll technician is prompted to re-enter login details or contact the system administrator
5. Payroll technician clicks on 'create a new account'
6. Payroll technician is redirected to a form page to enter employee details – first name, surname, pps no. etc.
7. Payroll technician enters new employee's details and clicks 'save'
8. PMS (Parsley Management System) creates a new sequential employee number
9. Payroll technician is prompted 'Send approval notification' and clicks 'yes'
10. PMS sends notification to System Admin
11. Prompt screen displays 'Waiting for Approval'
12. Payroll technician selects 'OK'

System Administrator

Features

The system administrator should have access, add and edit privileges for most areas of the system.

- Access features:
  Staff
  - ➢ Personal information
  - ➢ Bank details
  - ➢ Tax information
- Add/Edit Features:
  Staff
  - ➢ Rosters
  - ➢ Personnel file
  - ➢ Hourly rates/salary
  - ➢ Commission rates – if applicable
  - ➢ Targets – sales/project completion/performance
  - ➢ Performance records – achievements, punctuality, previous performance reviews
  - ➢ Holiday payments
  - ➢ Illness payments
- Specialised Requirements:
  - ➢ Notification feature – for holidays, illness, overtime etc.
  - ➢ Approval feature – to approve new accounts, account allocation to departments, payment for holidays/illness etc.
  - ➢ Install system updates
  - ➢ Resolve issues reported by other users

Use Cases

Example: System administrator logs into system to approve a new staff account

1. System administrator opens software to homepage
2. System administrator is prompted to login
3. System administrator r enters employee number/username and password
4. Outcomes:
   - ➢ Login attempt successful - System administrator is brought to the system administrator specific interface
   - ➢ Login attempt unsuccessful – System administrator is prompted to re-enter login details or contact the system administrator
5. System administrator is prompted ‘You have new notifications click here to view’
6. System administrator clicks the link and is redirected to the notification page
7. System administrator selects the notification
8. Notification says, ‘A new employee account is awaiting approval’
9. System administrator clicks on ‘Show details’ link and is redirected to the employee details page.
10. System administrator reviews new employee details
11. Outcomes:
    Approved
    - ➢ System administrator clicks ‘Approved’
    - ➢ System administrator is prompted ‘Activate the account’
    - ➢ System administrator clicks yes
    - ➢ PMS marks the account active and sends the new user his/her new password via e-mail

    Not Approved

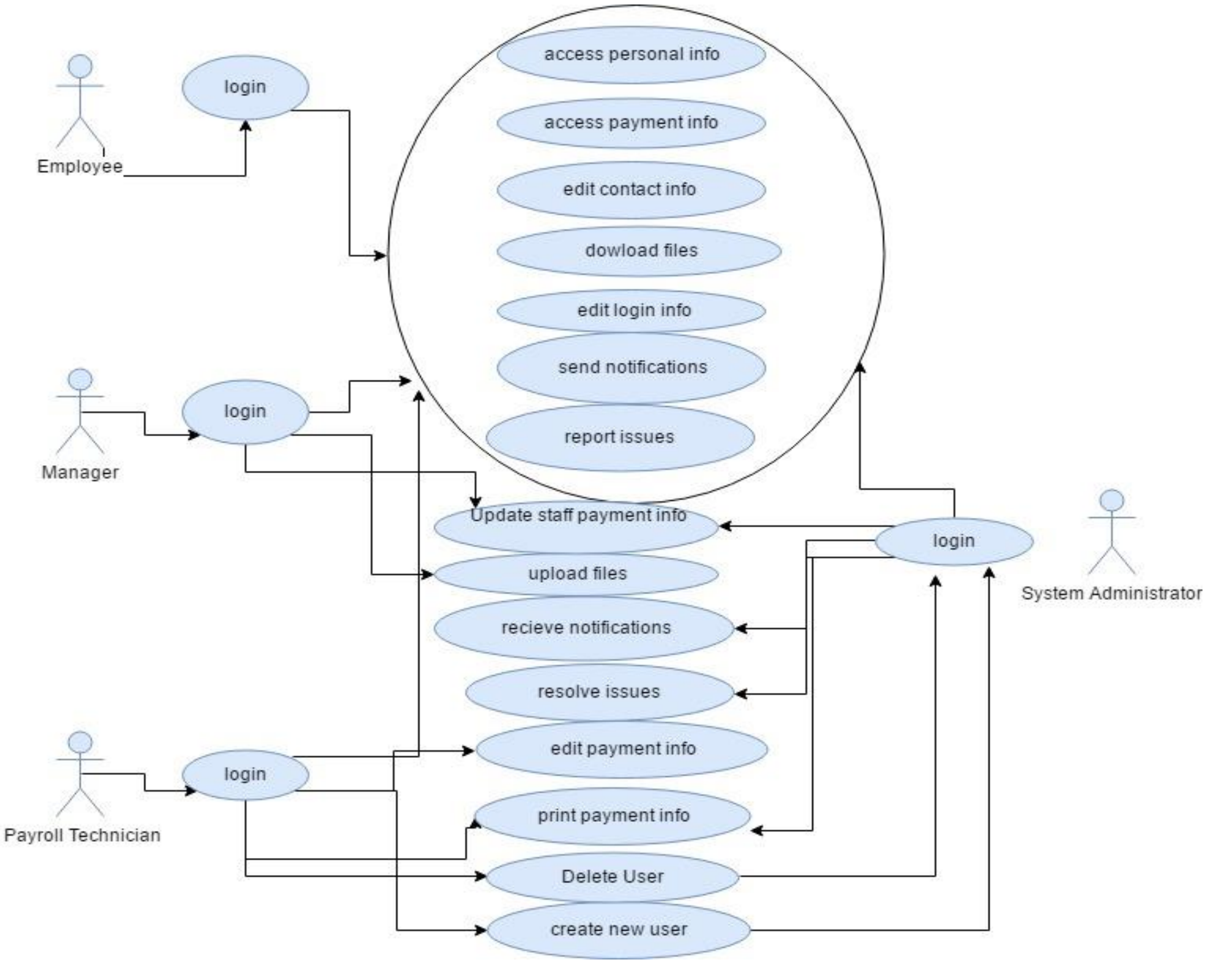
    - ➢ System administrator clicks ‘Not Approved’
    - ➢ PMS send notification to Payroll technician

- ➢ System administrator is redirected to notifications page

Example: System administrator logs into system to increase an employee's hourly rate

1. System administrator opens software to homepage
2. System administrator is prompted to login
3. System administrator r enters employee number/username and password
4. Outcomes:
    - ➢ Login attempt successful - System administrator is brought to the system administrator specific interface
    - ➢ Login attempt unsuccessful – System administrator is prompted to re-enter login details or contact the system administrator
5. System administrator clicks on 'Department' tab which displays options
6. System administrator clicks on 'Employee Details' tab
7. System administrator selects specific employee from a list
8. System administrator clicks on 'edit payment information'
9. System administrator selects 'Hourly Rate' from a list of options
10. System administrator is prompted to enter a new hourly rate
11. System administrator enters new rate and selects 'save'
12. System administrator is prompted again to review new rate and clicks 'Proceed'
13. System administrator is redirected to this employee's details
14. PMS sends notification to Payroll technician

Use Case Diagram (created at draw.io)



Threats

The Parsley Management System is a multi-user system and has many features which require user input, this is the biggest threat to the system as every user data entry point in a system is a possible way to exploit the system. I have outlined some of the more ominous and/or obvious vulnerabilities to the system which can be limited through deliberate programming decisions.

Access to Protected Data

All employee information must be stored in a database somewhere, like an SQL database for example. If an attacker wants access to private information currently stored in the database, the most common way to get this information is through an injection of malicious code. The attacker would navigate to an area where they can add or edit the database information, like changing contact information and when prompted to enter information, they would enter a statement in the place of their personal information in order to print the information they want.

Erasing Important Company Data

Following on from the previous threat, valuable data could be delete in much the same way, using a malicious code injection to wipe a database clean of the data stored there. This could be disastrous for a company depending on the data which has been erased.

Uploading Malicious Software

Any entry point for user data is a possible threat this way however as user data is easily limited to a certain number of characters, the most obvious point of entry for malicious software to be uploaded by an employee is in the manager account, where the manager can upload files for scheduling. Allowing entire files to be uploaded in a system gives an attacker a much easier way to infect the system with a malicious script/virus/malware which could be detrimental to the system.

Creating New Accounts

Allowing users to create a new account on the system is risky because if an attacker wants access to the system this is another way to do this. This could be internal – a payroll technician, external – a hacker, or a combination – a staff member helping an attacker. There are a few ways this could go wrong. There is a possibility of creating ghost employees, where wages are paid to an imaginary employee with a fake account. There is also a possibility that the fake account could be used for malicious reasons with less traceability. This type of attack would be less common in small companies where the system administrator and payroll technicians would know all the staff.

High-Level Privileges

Each role defined in the system has a unique range of permissions, two of the account levels, however, have high-level privileges which an attacker could use to completely destroy the system, the payroll technician has access to the most personal employee data and can edit employee payment information and approve payments. The System administrator has access to most areas of the system, including sensitive data and has the features required for the highest level of approval. An attacker gaining access to these interfaces could have serious consequences to the company.

Internal Threat – Disgruntled Employee

The threat of a malicious user is probably the most obvious for this system, simply because an account is required to even access the homepage. As each employee has add/entry features it is possible to initiate an attack from any account. Depending on the employee's intentions there are a number of threats to be considered, including:

Abuse Cases

Example: Employee gains access to another employee's contact information

1. Employee opens software to homepage
2. Employee is prompted to login
3. Employee enters employee number/username and password
4. Outcomes:
    - ➢ Login attempt successful - Employee is brought to the employee specific interface
    - ➢ Login attempt unsuccessful – Employee is prompted to re-enter login details or contact the system administrator
5. Employee clicks on the 'Contact Details' tab
6. Employee clicks on the 'Update contact information' tab
7. Employee clicks on 'E-mail address' tab
8. Employee is re-directed to a form page and prompted to enter new e-mail address
9. Employee enters malicious code to print entire database (eg.. OR1=1)
10. Database prints to screen
11. Employee searches through printed database to find the employee he/he is looking for
12. Employee records the information and logs out

Example: Manager uploads malicious script to system

1. Manager searches the web and finds malicious malware script
2. Manager saves script, named schedule.xls (Excel file)
3. Manager opens software to homepage
4. Manager is prompted to login
5. Manager enters employee number/username and password
6. Outcomes:
    - ➢ Login attempt successful - Manager is brought to the employee specific interface
    - ➢ Login attempt unsuccessful – Manager is prompted to re-enter login details or contact the system administrator
7. Manager clicks on 'Roster' tab which displays options- edit roster, delete roster, add new roster
8. Manager selects 'add new roster' from options
9. Manager is prompted to add file path or browse the system
10. Manager selects browse and selects the file on the computer
11. Manager selects upload
12. Outcomes:
13. File uploads successfully – Manager selects save file
14. File upload unsuccessful – Manager is prompted to try again or contact the system administrator
15. Manager is prompted 'make file available to view?'
16. Manager clicks 'yes' or 'save for later'
17. Malicious script is uploaded and processed by the system

External Threat – System Hacker

There are also threats to consider outside company employees, and because access to the system requires a login combination the only way to gain access to the system is to somehow acquire login credentials. This can be accomplished a number of ways:

1. Staff negligence

Staff negligence could lead to an attack on the system a number of ways. Staff giving someone their password for any reason is a security breach, also creating basic, easily guessed passwords like passw0rd, 12345678 etc., or writing down their login credentials, in their phone or beside their computer all leave the system vulnerable.

2. Brute Force Password Attack

This is when a hacker guesses password combinations, starting with the easy to guess passwords, until they find the correct combinations. There is software available for this type of attack, which cycles through key combinations systematically.

3. Dictionary Attack

A dictionary attack is similar to a brute force attack, the exception being a dictionary attack focuses on words and word combinations, beginning with the most common.

4. Key logger Attack

A key logger is a tool used by hackers to store keystrokes on a device. It can be both hardware and software and can capture password keystrokes, allowing an attacker to hijack an existing account.

Abuse Cases

Once an attacker gains access to an account the risk is the same as an internal attack

Example: Attacker deletes department data

1. Manager writes login information on a post-it and leaves it under the key board
2. Attacker is at manager's desk and sees login details
3. Attacker opens software to homepage
4. Attacker is prompted to login
5. Attacker enters employee number/username and password
6. Outcomes:
   - Login attempt successful – Attacker is brought to the employee specific interface
   - Login attempt unsuccessful – Attacker is prompted to re-enter login details or contact the system administrator
7. Attacker clicks on the 'Contact Details' tab
8. Attacker clicks on the 'Update contact information' tab
9. Attacker clicks on 'E-mail address' tab
10. Attacker is re-directed to a form page and prompted to enter new e-mail address
11. Attacker enters malicious code to delete entire database (eg.. 'DROP TABLE')
12. Database is deleted
13. Attacker records the information and logs out

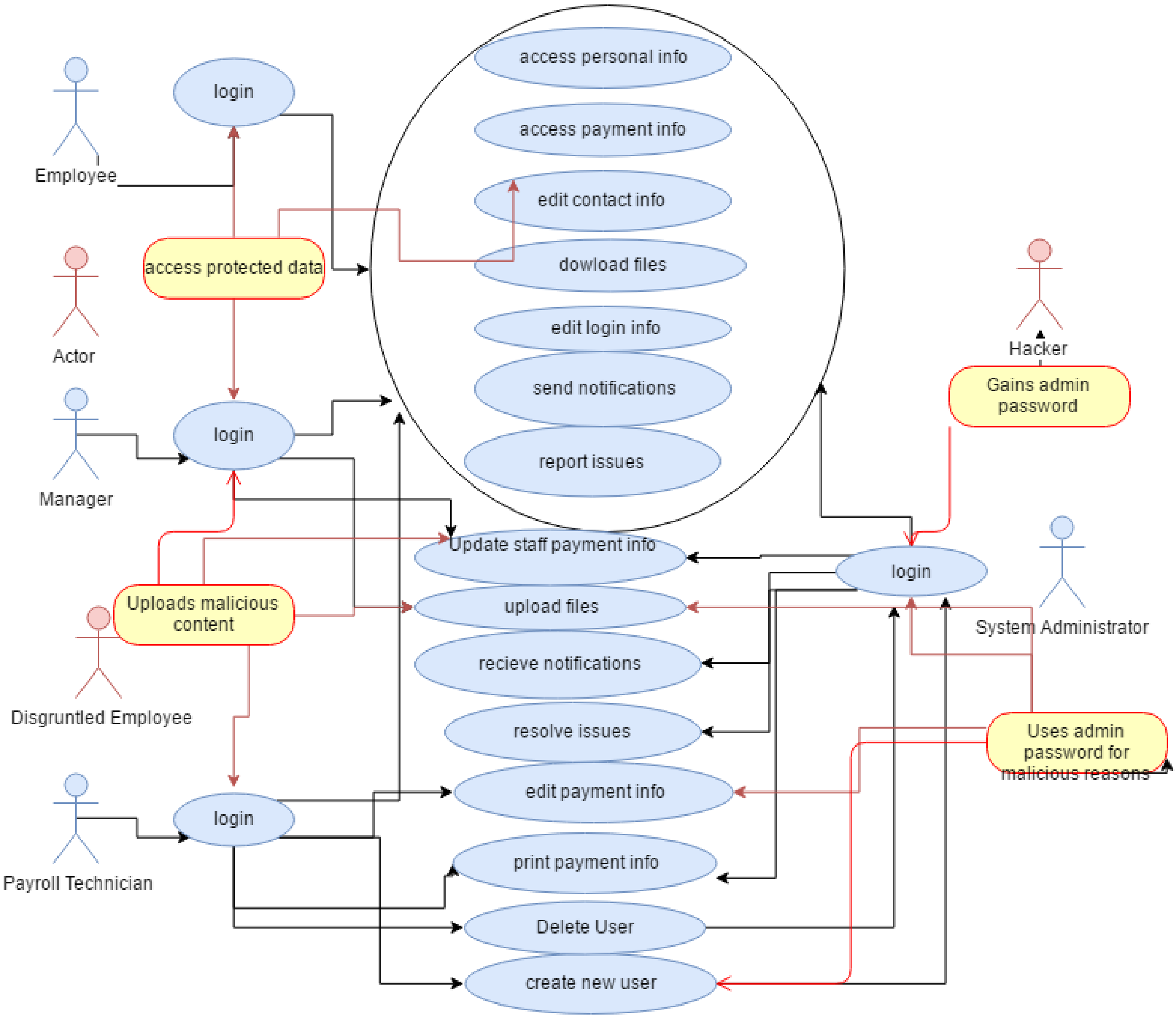
Example: Attacker edits employees' payment information

1. Attacker sends e-mail to system administrator
2. System administrator opens e-mail and click on link
3. System administrator's device installs key logging software
4. Attacker records system administrator's password
5. Attacker opens software to homepage
6. Attacker is prompted to login
7. Attacker enters employee number/username and password
8. Outcomes:
   - Login attempt successful - Attacker is brought to the employee specific interface

- Login attempt unsuccessful – Attacker is prompted to re-enter login details or contact the system administrator

9. Attacker clicks on ‘Department’ tab which displays options
10. Attacker clicks on ‘Employee Details’ tab
11. Attacker selects specific employee from a list
12. Attacker clicks on ‘edit payment information’
13. Attacker selects ‘Hourly Rate’ from a list of options
14. Attacker is prompted to enter a new hourly rate
15. Attacker enters new rate and selects ‘save’
16. Attacker is prompted again to review new rate and clicks ‘Proceed’
17. Attacker is redirected to this employee’s details
18. Attacker continues to edit payment information for all employees

Misuse Case Diagram (created at draw.io)



Threat Mitigation

Not all but some of these threats can be mitigated through the use of secure programming techniques, incorporating them into this system will drastically reduce attacks on the system. Some of the appropriate ways to do this are:

Validation

- Shorten the length of input fields
- Prevent the use of special characters
- Use regex validation to allow certain ASCII characters
- Use in-built functions for safety (e.g. HTMLSpecialChars ())

Use Secure Methods/Libraries

- Avoid libraries, or functions which are known to be insecure (e.g. C's get function)
- Avoid using dangerous system functions (e.g. shell_exec)
- If unsure of a frequently used method, research
- Follow best-practice guidelines in regard to functions use
- If an unsecure method is required, increase validation

Educate users

The company should create a strong password policy and increase awareness on the safety issue. Password policy should include:

- At least 8 characters
- At least one uppercase
- At least one lowercase
- At least one number
- No easily guessed passwords – passw0rd, 12345678 etc.
- No personal items – kids' names. Birthday dates etc
- No full dictionary words
- No writing down your password
- Don't tell anyone your password

Staff should also be educated and be aware of virus' and malware if accessing the system from personal devices.

Two Signature Authentication for Important

Two signature authentication is important for reducing fraud within a company but can also help reduce the damage from an attack. What I mean by two signature authentications is if one user makes a change, another user should have to authorise that change before it becomes permanent. This is included this in some areas of the system, however increasing this would significantly reduce the risk to the system.

Persistent Testing and Regular Updates

Persistent testing increases the awareness of security issues within the program. Allowing the program to be tested by a wide range of people increases the chances of vulnerabilities to be found unintentionally, and increases the scope of testing to multiple vantage points. Once an issue is found by testers, or reported by users an update patch can be released to improve the system and make it more robust.