Defeating shoulder surfing attacks with AI

6510 Final Project Writeup

Edward Crowder
Department of Computer Science
University of Guelph
Guelph, ON, Canada
crowdere@uoguelph.ca

Ahmad Chaiban
Department of Computer Science
University of Guelph
Guelph, ON, Canada
achaiban@uoguelph.ca

ABSTRACT

As remote work is on the rise, preventing unauthorized information disclosure through social engineering attacks, specifically shoulder surfing, becomes harder to detect and prevent. We propose a novel system utilizing the endpoints built-in webcam to detect and mitigate shoulder surfing through open-source machine learning models and a dynamic risk framework.

CCS CONCEPTS

Security and privacy ~ Social aspects of security and privacy

KEYWORDS

Security, Artificial Intelligence, Cloud, Social Engineering, Shoulder surfing

ACM Reference format:

Edward Crowder and Ahmad Chaiban. 2020. Defeating shoulder surfing attacks with AI: 6510 Final Project writeup. In *Proceedings of University of Guelph (University of Guelph'20), Guelph, Ontario, Canada 6 pages.*

1 Introduction

Shoulder surfing is a social engineering attack designed to gain unauthorized access to information by physically looking over someone's shoulder [1]. Historically, requiring workers to appear physically at a secure central office was a solution; however, Due to an increasing desire to work remotely and COVID-19 work from home (WFH) initiatives, the risk of information leakage through unauthorized individuals has increased. Information leakage can happen in many ways, intentional or unintentional. For example, a prying eye at a coffee shop might notice a slide with a new company acquisition, or someone at home might notice confidential information left open and gossip to a friend. Regardless of the situation, unauthorized access to confidential information poses a threat to the company and the individual who caused the leak.

The solution we refer to as "Enterprise Shield" uses multiple techniques and mechanisms to identify and authenticate people through the user's built-in webcam. Enterprise Shield (ES) can manipulate the user's system from desktop experience to corporate roles to maintain confidentiality from outsiders. Authentication through facial recognition paired with the currently logged in users profiles it is possible to cross-reference with a centralized remote active directory lookup. Identifying potential threats is as simple as the inverse of that, or the lack of records in the active directory.

2 System Design

Social engineering attacks such as shoulder surfing are considered to be physical attacks on enterprise information. The following system outlines a software-based approach to defending against unknown threat actors.

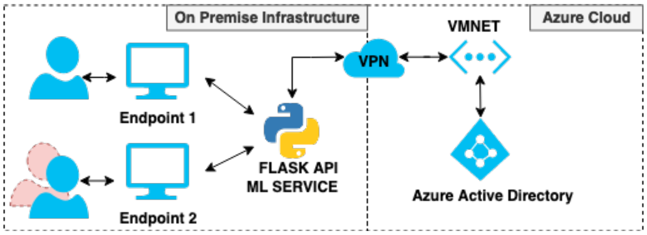


Figure 2.1 high level process

There are three major components to the system—first, a python-based endpoint client that feeds data to the risk analysis pipeline. Second, the AI models that evaluate our camera data, and third the Azure connector that enforces and authorizes security mechanisms. Currently, the solution is designed only for Mac; however, porting to Linux and Windows would be a reasonably simple task.

The endpoint client is responsible for processing the information provided by an active directory containing employee usernames and profile pictures. The client would be installed on every endpoint device within an organization to evaluate each employee's surrounding area. The goal is to identify unknown persons in the vicinity and ensure security mechanisms are triggered appropriately. Triggering appropriate security mechanisms is achieved by evaluating a risk score in real-time through the use of many open-source machine learning models. The models assess each frame of a scene to provide a count of entities identified, then map the count to a risk index to generate a score. If the score surpasses defined levels, consequence actions are triggered to protect the employee from unauthorized information disclosure.

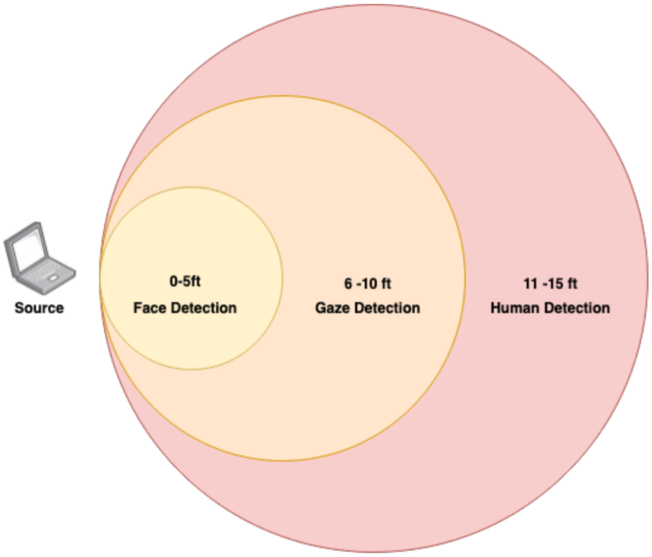


Figure 2.2 Defense in depth visualized from machine learning standpoint

By breaking down what shoulder surfing is and measuring the risks associated with its sub-patterns such as gaze, face, and being generally within view (human detection), it is possible to assign risk scores. We could control and minimize the disruption to work while maintaining integrity by implementing risk scores to patterns identified in shoulder surfing.

The choice for azure active direction is crucial
because Microsoft's active directory is popular among many large corporations. Azure Active Directory (AAD) makes the entire process accessible via an exposed REST API endpoint URL. Azure's cloud Active Directories can be easily merged with an on-premise Active Directory, making it a suitable choice to maximize flexibility. Organizations wishing to mitigate shoulder surfing attacks via software-level protections should deploy endpoint software as they already do to begin use. Furthermore, AAD endpoints allow us to change a user's active directory groups and permissions dynamically in real-time while also providing a canonical source for authentication.

3 Artificial Intelligence

The primary tools used to classify the different instances of shoulder surfing were some popular pre-trained object detection Machine Learning (ML) models. The objective was to identify any actors in a laptop camera's field of view. This involved discovering who (inside or outside an organization/enterprise) is looking at the given laptop, what they are looking at, and how many actors exist in the background.

In order to discover who was looking at the laptop, two methods were employed. **YOLO** Object Detection and **dlib** based face recognition. The former is used in order to detect an actor who is not in range of the face recognition system, and the latter is used up to a certain range in order to classify the face (by name) of an authorized/unauthorized employee, or an unknown actor altogether.

Regarding what the actor might be looking at, another **dlib and Computer Vision** based system was used in order to track an actor's eyes within a certain reasonable range. **If an actor's eyes are detected, it is assumed that the device is in danger of being compromised and the user's risk score increases**.

The entire system comes together in order to assess the risk level after processing every frame. Depending on the situations described in **reference**, the employee's laptop will receive a risk score.

3.1 Face Recognition (dlib & Computer Vision)

This face recognition model was taken from a library built with **dlib**, a C++ library that allows for the usage of various ML algorithms. It is 99.38% accurate and was benchmarked against the *labels of the wild* benchmark, a data set that is used to grade face recognition models.

The library provides various tools, such as facial outlining, feature extraction and real time face recognition. For the purposes of this defense system, real time face recognition was the only tool required. On every frame, the model would attempt to discover all the faces in range and either recognize an employee, which the system then checks their authorization, or if they're an unknown actor. Moreover, the model requires only one photo in order to train per face. This helps in both preserving employee privacy and allowing the system to train only on the Active Directory photos that already exist. Figure 3.1 provides an example of the detection being made by this AI system.

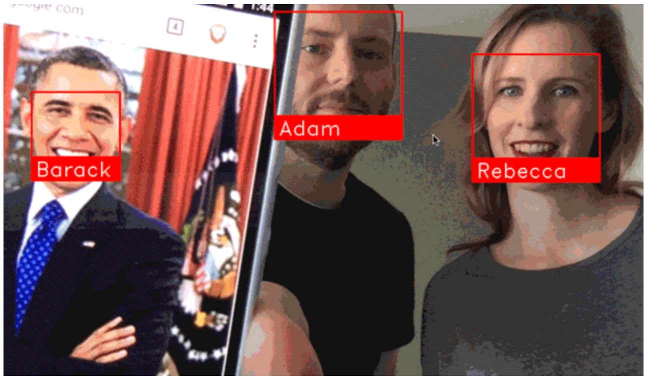


Figure 3.1.1: Face Recognition example

The model has two vulnerabilities. The first is that, as show in figure 3.1, photos can be used to trick the system. This can be protected against but is left in section 7 future works. One should also note that this model tends to make errors with children. This is a possible compromise if the person willing to commit a shoulder surfing attack wishes to send children to do their bidding.

3.3 YOLOv3-416 Human Detection

This model involves the famous YOLO detection model (You Only Look Once) that was trained on the COCO Dataset. Normally this model predicts 20 classes, but for our system, we only required it to predict one class, humans. Only classifications on humans were extracted from the model results, however, the model was not retrained, as that was thought unnecessary, the human predictions were sliced out of the default prediction array.

It is quite a robust model, and the detection is fast and accurate, able to predict humans obscured by certain obstacles, or far from the camera. It holds a mAP score of 55.3 and can operate at 35 fps. A pre-trained weights file is provided by DarkNet for open-source usage and was used for this system.

Human detection in this system is essentially used to determine the type of environment the employee is located in. If they are in a cafe, they probably are not in an environment that is secure for their work. And in this current system, the number of human's detected in the background might easily rack up the employee's risk score and consequently, the security measures.

Figure 3.2 provides an example of the result of YOLOv3-416 making a detection on an image containing humans.

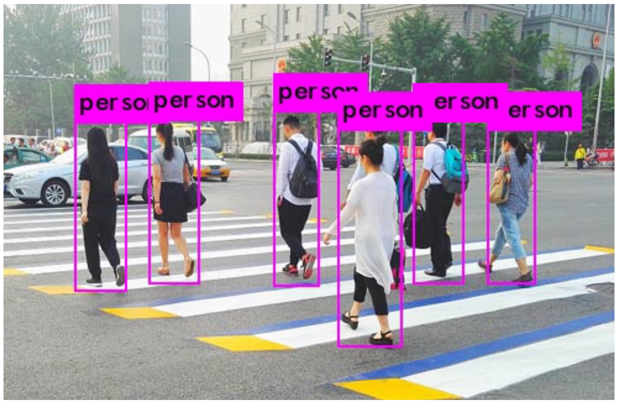


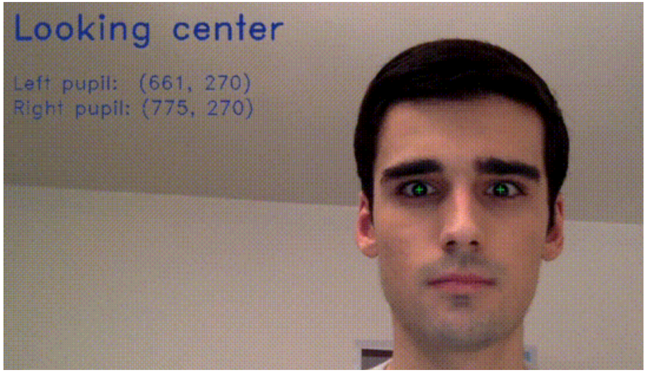
Figure 3.2.1: YOLOv3-416 Human Detection example

3.3 Gaze Detection

This model is also built on top of dlib and was an open source project by "antoinelame". In his repo, he outlined his usage of dlib and computer vision in order to make a frontal face detection, estimate the location of the eyes, pupils and track their subsequent movements.

It is quite accurate, as it uses pre-existing methods in a well-rounded AI system, and gives resulting ratios and precise coordinates on the different elements of the eyes. Moreover, these calculations are used in order to determine a where the actor's gaze is set on through various thresholds. Figure 3.3 provides an example of the gaze tracking detection. The pupils in this case are highlighted in green and some coordinates and additional information is given regarding the frame in question.

It should be noted that some changes were incorporated into this particular AI system's architecture. First, the model initially only supported the detection of one pair of eyes, however, it had to be enhanced in order to detect multiple pairs of eyes. Also, the system had certain limitations. If one of the actor's eyes were not present it would not make a successful detection. This had to be rectified by eliminating the limitation altogether.


```
Looking center

Left pupil:  (661, 270)
Right pupil: (775, 270)
```

Figure 3.3.1: anotoinelame's Gaze tracking system

3.4 Other Tested Models

Some other models were tested before these final three were selected. The first was a purely computer vision based eye detection and tracking model. However, it was not robust enough and did not provide sufficient details to make a conclusion regarding an actor's gaze. It was therefore not considered as part of this system.

Another model was Facebook's famous Detectron2, however, it did not perform well on a standard CPU based system, and therefore as part of this system which we consider to be scalable and affordable, it overworks the system for many kinds of detections and perhaps required retraining that was unnecessary. The human and face detection combination that was finally chosen was sufficient.

A final model that was considered for this system was also a Computer Vision based human detection model. However, its detections were extremely sensitive to light and distance. It was not robust enough and was not considered to be part of this system.

The selected models were finally combined in order to annotate each individual frame with Computer Vision and render a result in order to verify the integrity of the Machine Learning in the system. However, these are not to be included in the final product, as it is solely for developer usage and would put unnecessary processing on an enterprise's system.

4 Security Mechanisms

Security mechanisms to prevent unauthorized individuals from gaining access are critical to the success of this project. We have implemented many security features that can be triggered based on a customizable risk score. The following section breaks down our implemented security mechanisms and outlines their role in the larger system.

4.1 Correlating user to profile

The ability to get the username of the currently signed-in user is the most critical feature. Identifying the current user is crucial because it allows us to compare the facial recognition output to the individual using the machine. The username collected is compared to their Azure Active Directory (AAD) profile, and every other security mechanism feed from this. The AAD profile will include by default information such as corporate username, job title, department, manager, and profile picture. A customizable risk score is applied to each threat and determines what actions are triggered. The risk increases determined on a customizable threat model described in section 5 Risk Score.

4.2 System notifications

The first security mechanism we implemented was to generate a notification on the system. We wanted to notify the user of potential threats in their background without interfering with their work. We assumed that it was reasonable that an employee-focused on their task at hand may not be fully aware of their surroundings and therefore might not notice someone behind them immediately. Sending a default system notification allows us to prompt the employee with a visual and audible cue. By default, this notification will trigger based on any unidentifiable humans in the background and hopefully encourage the employee to move to a more secluded place to work.

4.3 Hide / Show windows

In the use case that we have identified faces that are not in the employee database, we are at risk of information disclosure. Our solution is to hide the active windows on the screen until the possible threat is no longer in view. This use case can also apply to members of the employee database in other departments or even members of the same department but in different positions. For example, in most organizations, it would be unacceptable for anyone outside human resources to be looking at the human resource screen, similarly with the finance department regardless of industry. The future work explains a robust system to configure this hierarchy of roles in correlation to your AAD configuration.

4.4 Locking the device

Locking the screen and requiring them to re-authenticate themselves causes the employee to be mindful of their location and reorient themselves to reduce overall risk. There are many use cases for when we might want to lock the employee's machine, such as:

- Employee is not in view
- Face detection detects unknown individuals in the scene
- Too many people in the background
- Etc.

Ultimately, the employee will not be able to continue their work until they relocate or make an effort to reduce the number of unknown individuals in the background.

4.5 Limitations

The primary limitation for us to implement these security features would not be an issue in production environments, and therefore, it does not compromise the overall output of this project. The largest limitation surrounds the Azure cloud itself. Azure free tier does not allow us to create custom attributes within the AAD profile of an employee. To satisfy this requirement, upgrading to the Azure B2C level is necessary. Azure B2C comes at a considerable cost and is out of scope for our project; however, interfacing with it is just an extension of the current implementation. Section 7 FUTURE WORKS that covers the benefits and next steps connected to an upgraded Azure B2C subscription.

Overall, we have described a minimal subset of possible security mechanisms and use cases to protect against shoulder surfing. Through constant reminders such as system notifications and interference in their workflow, we hope employees become more aware and mindful of their working location.

5 Risk score

Now that there is a fundamental understanding of the artificial intelligence detection methods described in section {{2? AI}} and the security mechanisms described in section {{4? Security mechanism}} it is possible to calculate risk and take action based on consequence.

In this section, we have assigned risk scores to listed scenarios in order to demonstrate functionality. Table 2.1 outlines the base risk score index, and table 2.2 demonstrates the consequence associated with reaching each risk level. As risks are added or removed from the scene in real-time, the risk score will grow and shrink dynamically, always returning to its base value of zero when the currently signed in user match is the only user in the scene.

Detection	Risk score increase
Known Authorized Face	Reset to 0
Unknown human	+5
Gaze Detected	+10
Known Face (unauthorized)	+15
Unknown Face	+25

Table 2.1 Risk score index

Based on our scoring index in table 2.1, if an unknown human is detected in the scene, the risk score will be increased from zero (0) to five (5), and the employee will be notified, as displayed in table 2.2. These risk scores can be compounded such that if this unknown human is also detected to gaze at our screen, the score will be increased further, resulting in a total of 15 (Unknown human (5) + Gaze (10) = 15), causing the active windows to be hidden as well as the employee notified.

	Risk Score					
Defense	**0**	**5**	**15**	**25**	**50**	**75+**
No Action	🟦					
Notify User		🟦				
Hide Windows			🟦			
Lock screen				🟦		
Change Perm level[1]					🟦	
Block user sign-in[1]						🟦

Table 2.2 Risk score consequence correlation matrix

The scoring index has been calculated such that if a single unknown threat actor is simultaneously detected as all of the above, the device will lock the screen, causing the employee to deal with the current situation before signing back in. Since it is possible to have multiple instances of the same detection, it is also possible to have the device lock if there are five (5), unknown humans in the background. However, since facial detection is the only model with the concept of authorized or unauthorized, it is possible to detect an authorized face, unknown human, and unknown gaze; In this scenario, the facial recognition negates the risk score generated by the other detections.

6 Conclusion

Detecting and mitigating social engineering attack shoulder surfing proved successful through using multiple pre-trained machine learning models. Serving machine learning results through a flask service showed increased performance compared to locally evaluating risks while also improving the environment's overall security.

A trivial risk framework was implemented on the endpoint to trigger defense mechanisms such as generating notifications, hiding windows, locking the machine, and neutralizing permissions based on severity to protect confidential information. Although not all successful, our primary limiter was the Azure free tier active directory service. Azure free tier allowed us to connect to the user database for authentication and identification but does not allow custom fields; however, switching the tenant ID to a premium Azure B2C service would be all that's required to move forward with development.

7 Future work

The primary objective of this project was to identify known employees via facial detection and restrict information disclosure from shoulder surfing attacks. To this extent, we have succeeded, however there are many possible interdisciplinary next steps to take this software to production levels, such as:

- Implement configurable risk score and consequence table via same method as azure config.
- Implement azure B2C endpoints with custom attributes for central violation tracking
- Create a better formula for risk scoring (if required: super basic right now)
- design and implement logging and reporting for incidents
- Add photo detection (not a real person but a picture used to trick the camera such as figure 3.1) in order to protect against that
- Investigate privacy, ethical, moral, and legal implications
- Investigate human psychological implications regarding work productivity and being “monitored” constantly.

[1] Not currently implemented due to limitations

REFERENCES

[1] EC-Council, Unknown. “Shoulder Surfing: More Prevalent than You Would Imagine: EC-Council Official Blog.” Ethical Council - Blog, 1 Nov. 2019, blog.eccouncil.org/shoulder-surfing-more-prevalent-than-you-would-imagine/.

[2] Gonçalves, Luana. “Detecting People with YOLO and OpenCV.” Medium, Medium, 24 Nov. 2019, medium.com/@luanaebio/detecting-people-with-yolo-and-opencv-5c1f9bc6a810.

[3] Kamal, Amro. “YOLO, YOLOv2 and YOLOv3: All You Want to Know.” Medium, Medium, 28 Oct. 2020, medium.com/@amrokamal_47691/yolo-yolov2-and-yolov3-all-you-want-to-know-7e3e92dc4899.

[4] Unknown, Pascal. “YOLOv3 Trainde on VOC2007(Person Class) Failed to Detect Small Objects , Anchors Are Changed but Have No Effect · Issue #972 · Pjreddie/Darknet.” GitHub, 2018, github.com/pjreddie/darknet/issues/972.