# PERSONASCOPE: Defending Against Persona Abuse Attacks

Rajiv Sarvepalli

University of Virginia – Department of Computer Science

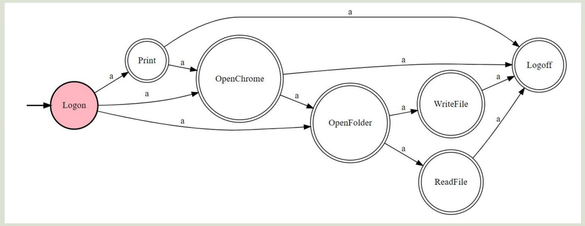
## INTRODUCTION

The influence of advanced cyber attacks (e.g., Advanced Persistent Threat) is ever-increasing. Notoriously, these attacks frequently use stolen user credentials to frame innocent users for the attacker's crimes. While attempts have been made to detect abnormalities in user behavior to identify real users behind explicit credentials (e.g., log-in user names), the current modeling of user behavior is limited. Simply detecting anomalies in user actions is insufficient; users are too dynamic and often change their behavior (as they change their roles).

To solve this problem, we develop an advanced user modeling technique that automatically synthesizes realistic user models to identify (ideally all) possible user models including those of attackers. Specifically, we formulate the problem as a searching problem for user models within the complete user model space which includes all the plausible users made from combinations of user actions (i.e., reading a file, printing, opening an application) based on states.

## OBJECTIVES

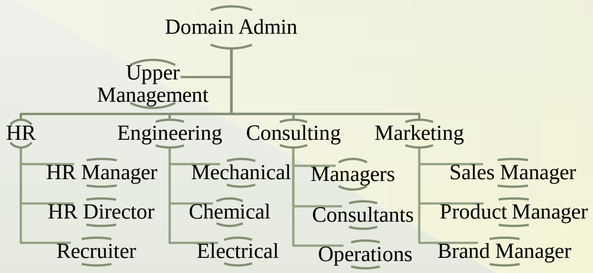
- Create a user modeling system that has models for every plausible user
- Synthesize user behavior as a baseline through defining user actions
- Create definitions of users through the synthetic user behavior
- Match real world users to the definitions of users in order to track alterations of their behavior

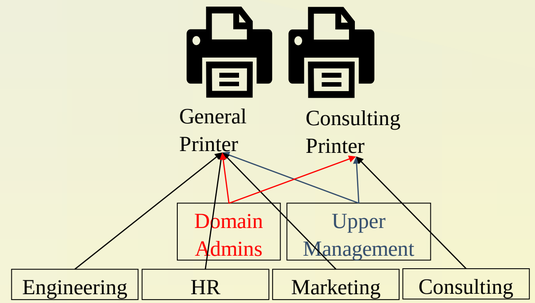



User behavior can be modeled through the creation of a finite state machine where the states represent actions and the transitions are user choices. A complete traversal of a finite state machine would allow for the modeling of all possible users. This would allow for the synthetic baseline to be complete and guarantee any real user to have some synthetic user it matches with. Above is a simple example of this where "a" represents a user choice. Repeated actions could simply be illustrated with a loop.

## MATERIALS AND METHODS

We planned a realistic mock environment (similar to a real corporation) containing 10 different virtual machines that are apart of the virtual domain. The virtual domain was set up to mimic that of a corporate environment with separations of department and differing levels of control and access. The hierarchy within the virtual domain looked something like the graphic below on the left. Below on the right is an illustration of shared printer access where an arrow represents access to that printer.



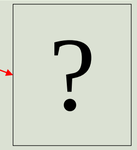

These computers were executed to perform basic user actions under different domain users and the logs were collected on these results. These actions were composed by creating a basic list of user actions and then chaining them together in a sensible manner to form one synthetic user. Initially, the actions were defined with dependencies and results, and actions were placed in order such that the summed results of the previous actions would be greater than or equal to the dependencies of the current action. However, this plan only leads to one set of actions (one user) and does not include the fact that actions can be repeated. By using a finite state machine, actions could be linked together to create an overall picture of all the possible users. Actions were linked with the finite state machine using the earlier model except actions could repeat themselves. Below is an example of how a set of actions can be made into a synthetic user and how matching real user would work. In the scenarios below the unknown individual would be considered an attacker since it does not match any of the known users.

| Synthetic User A |
|---|
| 1. Logon |
| 2. Open Chrome |
| 3. Read from Shared folder G |
| 4. Logoff |

| Synthetic User B |
|---|
| 1. Logon |
| 2. Read from Shared folder G |
| 3. Logoff |

| Synthetic User C |
|---|
| 1. Logon |
| 2. Print to Printer X |
| 3. Open command prompt |
| 4. Add a new domain user |
| 5. Logoff |

| Unknown Individual Y |
|---|
| 1. Logon |
| 2. Open command prompt |
| 3. Promote user to domain admin |
| 4. Logoff |

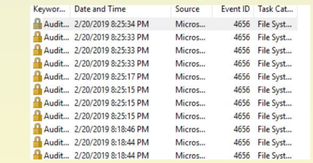



When traversing the finite state machine, a heuristic was added to ensure that actions that could be repeated would have less chance of doing so as they continued to be repeated to stop infinite actions being linked together. When a final traversal was complete, ideally, all possible users would be generated. In reality, the majority of possible users of that set of actions would be created.

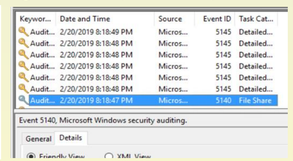
Using an open-source tool by MITRE called Caldera, the set of actions were able to be easily executed. MITRE's Caldera was created emulate attacker behavior after the initial break-in of an APT attack, but we modified it to be able to run all sorts of basic user actions we required. Once the user actions were able to be executed, the next step was to add attacker actions under the same domain account. That way we could collect the event logs and see if there was a noticeable change from a benign user behavior to an APT attacker abusing the innocent user's credentials.

## RESULTS

The generated actions that compose a synthetic user were executed and the logs from the domain controllers were connected. These logs attempt to emulate benign user behavior. An example collected log is shown (below right). Attacker actions emulated using MITRE's open-source Caldera were run independently and the logs from the domain controllers were connected. An example collected log is shown (below left).



| Keywor... | Date and Time | Source | Event ID | Task Cat... |
|---|---|---|---|---|
| Audit... | 2/20/2019 8:25:34 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:25:33 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:25:33 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:25:33 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:25:17 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:25:15 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:25:15 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:25:15 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:18:46 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:18:44 PM | Micros... | 4656 | File Syst... |
| Audit... | 2/20/2019 8:18:44 PM | Micros... | 4656 | File Syst... |



| Keywor... | Date and Time | Source | Event ID | Task Cat... |
|---|---|---|---|---|
| Audit... | 2/20/2019 8:18:49 PM | Micros... | 5145 | Detailed... |
| Audit... | 2/20/2019 8:18:48 PM | Micros... | 5145 | Detailed... |
| Audit... | 2/20/2019 8:18:48 PM | Micros... | 5145 | Detailed... |
| Audit... | 2/20/2019 8:18:48 PM | Micros... | 5145 | Detailed... |
| Audit... | 2/20/2019 8:18:48 PM | Micros... | 5145 | Detailed... |
| Audit... | 2/20/2019 8:18:48 PM | Micros... | 5145 | Detailed... |
| Audit... | 2/20/2019 8:18:47 PM | Micros... | 5140 | File Share |

Event 5140, Microsoft Windows security auditing.

General Details

## CONCLUSION AND FUTURE WORK

Persona abuse attacks are something that is becoming ever more widespread. By creating synthetic users and matching real users to those synthetic users, the transitions of user behavior should be noticeable since there are clearer boundaries between synthetic user and attackers when compared to real users and attackers.

Later, real users in a real-world business environment can be fitted to a user model, and a change of user behavior can be precisely detected as a transition to a different model. The categorization of users proposed above allows for accurate user identification enabling precise determination of user transitions. Anomaly detection of user transitions from one model to another will provide more precise detection of inappropriate user behaviors including malicious behaviors. We plan to release the code as open-source and plan to write a paper on the implementation and design details. Additionally, we will seek industrial partners to deploy the system in the future.

## REFERENCES

[1] Miller, D.; Alford, R.; Applebaum, A.; Foster, H.; Little, C.; and Strom, B. 2018. Automated adversary emulation: A case for planning and acting with unknowns

[2] A. Applebaum, D. Miller, B. Strom, C. Korban, R. Wolf, "Intelligent automated red team emulation", *Proceedings of the 32nd Annual Conference on Computer Security Applications. ACM*, pp. 363-373, 2016.

## ACKNOWLEDGMENTS

I thank Professor Yonghwi Kwon for useful discussions on implementation and experiment design.

This work was supported by the Semiconductor Research Corporation (SRC).