BCS Edinburgh branch

19:00- 20:00, 7 July 2021, Edinburgh, Scotland

Invited Talk:

Data Driven Mobile Endpoint Security

- Detecting Malware and Unauthorised Access

Zhiyuan Tan (PhD)
Associate Professor in Cybersecurity
School of Computing
Edinburgh Napier University, U.K.

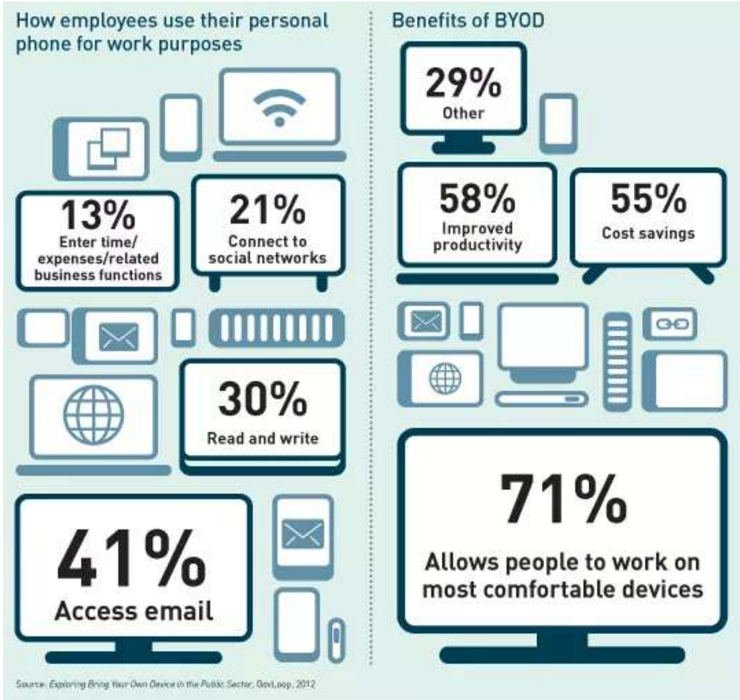
Napier UNIVERSITY

Agenda

- What is endpoint security?
- Mobile threats and vulnerabilities
- Challenges to securing mobile endpoint
- A bio-inspired malware analysis mechanism
- Touchscreen Input as a Behavioural Biometric for Continuous Authentication

What is Endpoint Security?

- The process of securing the various endpoints on a network, often defined as end-user devices such as
  - Mobile devices
  - Laptops
  - Desktop PCs



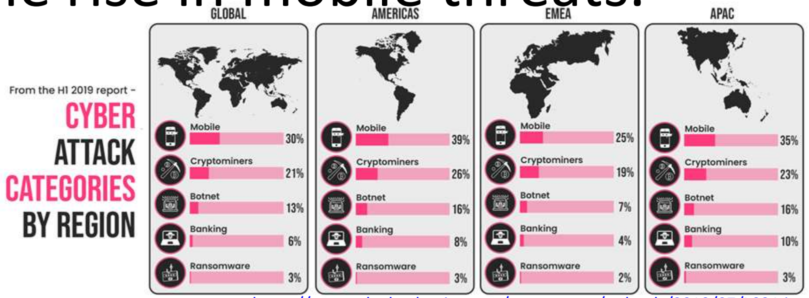
Endpoint Security is Increasingly Important

- The enterprise network security perimeter has essentially dissolved
  - As more enterprises adopt practices, such as BYOD and mobile employees
  - Mobile devices, such as mobile phones and tablets, are involved widely in our daily life activities
    - Banking
    - Shopping
    - Social networking
    - Education
    - Mobile working

The Case for Making BYOD Safe, 2015

Endpoint Security is Increasingly Important

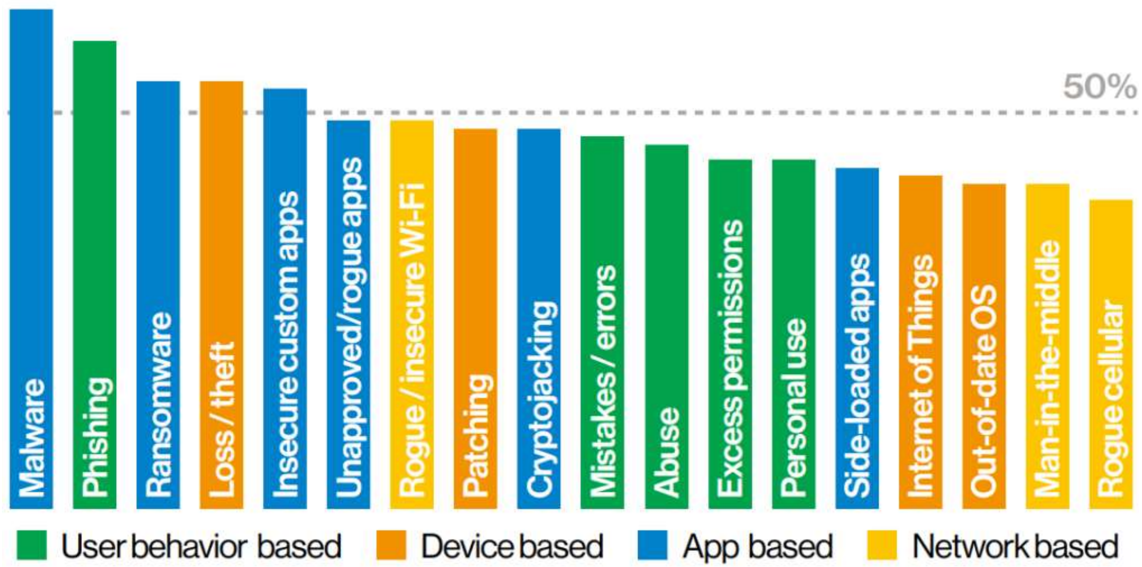
- In Q1 2020 alone, about 375 threats per minute with a 71% increase in mobile malware as compared to Q4 2019.
  - Particularly aimed at employees working from home due to the Covid-19
  - Nearly 1 million Covid-19 related malicious file detected affecting about 4,355 organisations
- The need for effective endpoint security measures has increased substantially, particularly in light of the rise in mobile threats.



https://research.checkpoint.com/wp-content/uploads/2019/07/L031.jpg

Mobile Threats and Vulnerabilities

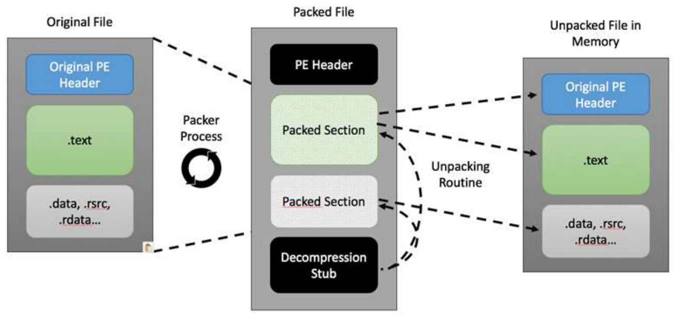
- Top four threats/vulnerabilities
  - **Malware**
  - Phishing
  - **Ransomware**
  - **Lose/theft**



https://businessinsights.bitdefender.com/hs-fs/hubfs/malware.png?width=1109&name=malware.png

Challenges to Securing Mobile Endpoint

- An overwhelming number of new malware
  - Very unlikely for human experts to catch up with
- Obfuscation techniques are employed to build malicious binaries to prevent detection
  - Malware packing
  - Polymorphism
  - Metamorphism



- Current analysis and detection techniques are mostly reactive and therefore cannot detect unseen attacks

Metamorphic Malware

- **A most challenging type** of malware that changes its entire code between generations
  - Usually, this involves the malware mutating itself and hiding its instruction within the normal program code of the host machine
- Metamorphic malware transform their codes using the following **mutation techniques**:
  - Instruction substitution
  - Garbage code insertion
  - Variable substitution
  - Control-flow alterations

Metamorphic Malware

- Metamorphic malware binaries have various layers in which they mutate
  - Network Layer Mutation
  - Application Layer Mutation
  - Exploit Layer Mutation
- The most challenging type of malware
  - The variants of metamorphic malware are very dissimilar
  - It often goes undetected as detectors are trained to recognise only specific code versions
- Nearly 100% new malicious programs discovered in recent years emanate from Android platforms

Metamorphic Malware Detection & Key Challenges

- The detection strategies
  - Signature-based detection
    - Challenges: recognise specific code versions and need to be fed with signatures of new potential variants
  - Heuristic-based detection
    - Challenges: suffer from the underlying vulnerability of Machine Learning (ML) algorithms; adversarial samples can be generated to evade detection
  - Malware normalisation and similarity-based detection
    - Challenges: more sophisticated transformations could seriously undermine the effectiveness of static analysis

Exploring for Solutions

- Our observations
  - Adversarial learning techniques
    - Deliberately feed them with malicious inputs (adversarial samples)
    - Discover loopholes and subsequently improve detection models
    - Make detection models more robust to attack
  - However, the lack of adversarial training data for ML models to learn from impedes model generality
  - Potential solutions:
    - **Evolutionary Algorithms** (EA) have shown effective in code optimisation (White, Arcuri, & Clark, 2011), (Langdon & Harman, 2015) and (Cody-Kenny, Lopez, & Barrett, 2015)

Generate New Malware Samples Using EA

- Our Proposition
  - Use EAs to create mutant variants of malware that are diverse and as evasive as their parent malware
  - Improve ML models by augmenting training data with the diverse set of evolved mutant samples created using the EAs
- Evolutionary Algorithm (EA)
  - A population-based meta-heuristic search process inspired by processes occurring in biological evolution that guides a population to adapt towards a desired goal
  - Huge search space for the exploration of code variants

Generate New Malware Samples Using EA

- Objectives:
  - To evolve new evasive variants of malware to be diverse enough so that they can represent a wide variety of potential states the malware can morph to
- Produce a well-informed fitness landscape
  - To discover new evasive variants that are significantly dissimilar from the original malware in terms of
    - Their code structure
    - Their behaviour
  - But to achieve the same goal

Generate New Malware Samples Using EA

- A novel Evolutionary Algorithm (MAL_EA) for malware variant generation
  - Employs a canonical model of an EA
  - Returns the single best mutant found (according to the fitness function) at the end of each run
  - Is run multiple times to get multiple variants
  - Features
    - Initialisation
    - Mutation Operators
    - Selection
    - Fitness functions: *Detection Rate*, *Behavioural Similarity*, *Structural Similarity*

Algorithm 1 Evolutionary Algorithm — MAL_EA

```
𝒫 ← initalise_random_population()          ▷ created by mutating original malware
evaluate(𝒫)                                 ▷ score each malware according to chosen fitness metric
f_w = min(𝒫)                                ▷ worst member of population
while Maximum number of iterations not reached do
    p ← select_parent(P)
    c ← mutate(p)
    f_c ← evaluate(c)
    if f_c > f_w then
        𝒫 ← 𝒫 ∪ c                           ▷ add child to population
        remove_worst()                      ▷ remove least fit mutant
        f_w = min(𝒫)
    end if
end whilereturn 𝒫
```

K. O. Babaagba, **Z. Tan**, and E. Hart, "Nowhere metamorphic malware can hide – a biological evolution inspired detection scheme," in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Influence of Fitness Function on Evasiveness

- Following 10 runs using each of the fitness functions
  - Count of malicious variants returned from 10 runs of the EA under each of the 3 fitness functions as shown in the table below.

Fitness Function	Dougalek	Droidkungfu	GGtracker
DR(x)	7	10	9
BS(x)	7	9	8
SS(x)	10	9	7

  - The percentage of detectors that fails to recognise the novel variants over the *x* runs that are malicious
    - Dougalek: Original-40.3%; DR(x)-72%*; BS(x)-66.7%*; SS(x)-67.3%*
    - Droidkungfu: Original-65%; DR(x)-94%*; BS(x)-82.1%*; SS(x)-83%*
    - Ggtracker: Original-38.3%; DR(x)-73.3%*; BS(x)-62.1%*; SS(x)-62.1%*

*The best evolved variants of the original malware

K. O. Babaagba, **Z. Tan**, and E. Hart, "Nowhere metamorphic malware can hide – a biological evolution inspired detection scheme," in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Influence of Fitness Function on Evasiveness

– Diversity w.r.t
  - Detection signature

	Dougalek	Droidkungfu	GGtracker
Detection	43	90	78
Behavioural Similarity	71	89	50
Structural Similarity	50	33	29

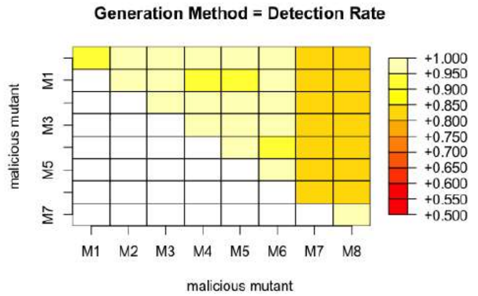
  - Behavioural signature

	Dougalek	Droidkungfu	GGtracker
Detection	100	70	89
Behavioural Similarity	100	78	78
Structural Similarity	80	75	100

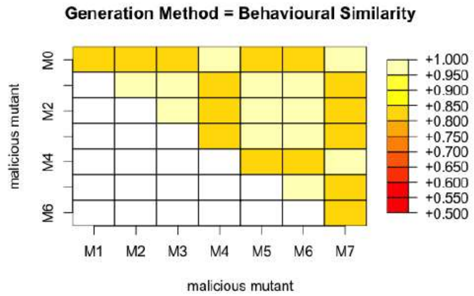
K. O. Babaagba, **Z. Tan**, and E. Hart, "Nowhere metamorphic malware can hide – a biological evolution inspired detection scheme," in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Influence of Fitness Function on Evasiveness

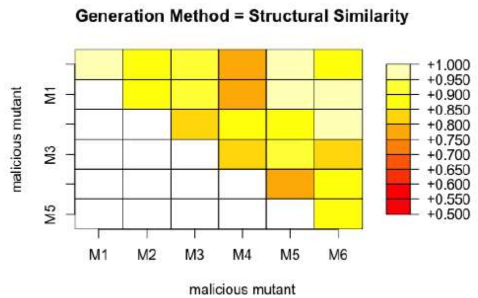
- Structural similarity - GGtracker



(a) Structural Diversity GGtracker for function DR(x)



(b) Structural Diversity GGtracker for function BS(x)

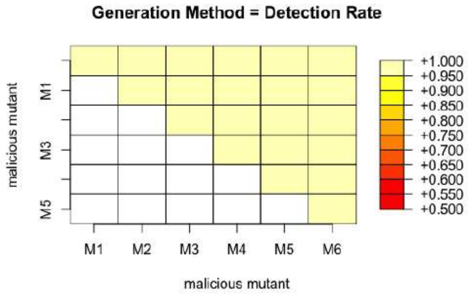


(c) Structural Diversity GGtracker for function SS(x)

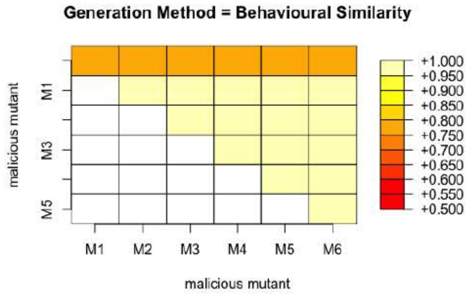
K. O. Babaagba, **Z. Tan**, and E. Hart, “Nowhere metamorphic malware can hide – a biological evolution inspired detection scheme,” in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Influence of Fitness Function on Evasiveness

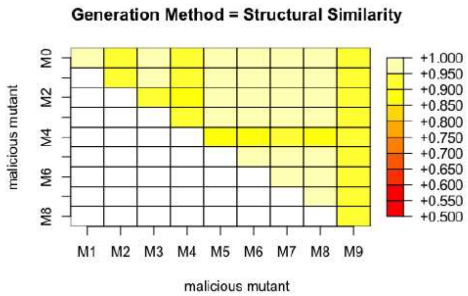
- Structural similarity - Dougalek



(a) Structural Diversity Dougalek for function DR(x)



(b) Structural Diversity Dougalek for function BS(x)

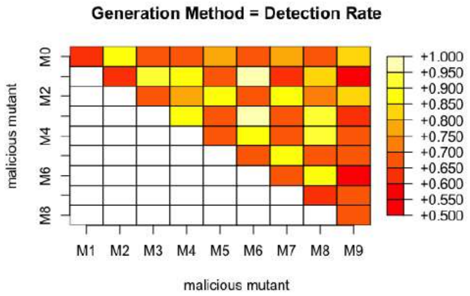


(c) Structural Diversity Dougalek for function SS(x)

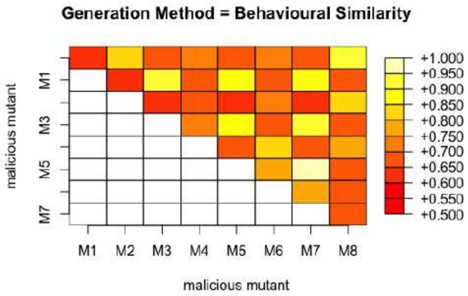
K. O. Babaagba, **Z. Tan**, and E. Hart, "Nowhere metamorphic malware can hide – a biological evolution inspired detection scheme," in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Influence of Fitness Function on Evasiveness

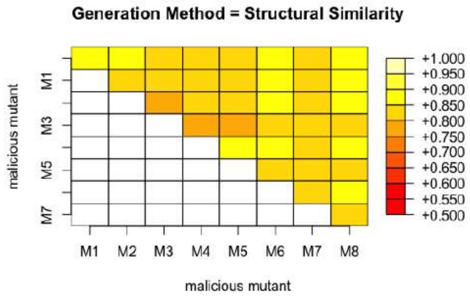
- Structural similarity - DroidKungfu



(a) Structural Diversity Droidkungfu for function DR(x)



(b) Structural Diversity Droidkungfu for function BS(x)



(c) Structural Diversity Droidkungfu for function SS(x)

K. O. Babaagba, **Z. Tan**, and E. Hart, "Nowhere metamorphic malware can hide – a biological evolution inspired detection scheme," in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Limitations of the EA-based Approach

- Limitations
  - Generate only a single new sample with each run of the algorithm
    - Time-consuming to generate a good enough number of samples
  - No guarantee that the samples will be diverse
  - Lack of insight into the properties of the generated samples
- Improvement
  - MAP-Elites (Mouret and Clune, 2015):
    - One of a new raft of Quality-Diversity optimisation algorithms that aim to return an archive of diverse, high-quality behaviours in a single run
    - Traverses a high-dimensional search space in search of the best solution at every point of a feature space, with low dimension defined by the user

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites, 2015.

Tailored MAP-Elites for Malware Mutant Generation

- Tailored MAP-Elites
  - A two-dimensional grid is
    - Defined by two features:
      - The behavioural similarity of a solution to the original malware
      - The structural similarity of a solution to the original malware
    - Divided in 20x20 equally sized cells that
      - are created by equally "binning" the range of each feature (which take values [0, 1])
      - Create a potential archive of 400 solutions
  - The algorithm is initialised with
    - A random population of mutants, each created by applying a single mutation to the original malware

Algorithm 3 MAP-Elites algorithm for mutant generation, modified from [88]

procedure MAP-ELITES(I, G)
  ($\mathscr{E} \leftarrow \phi, \mathscr{X} \leftarrow \phi$) ▷ N-dimensional map of elites: mutants $\mathscr{X}$ and their evasiveness $\mathscr{E}$
  for iter $= 1 \rightarrow I$ **do** ▷ Repeat for I iterations
    if iter $> G$ **then** ▷ Initialize by generating G random solutions by mutating original malware
      $x' \leftarrow random\_solution()$
    else ▷ Subsequent solutions are generated from elites in the map
      $x \leftarrow random\_selection(\mathscr{X})$ ▷ Randomly select an elite x from the map $\mathscr{X}$
      $x' \leftarrow random\_mutation(x)$ ▷ Create a mutant of x
    end if
    if $executable(x')$ **then** ▷ Confirm that mutated solution compiles and executes
      $b' \leftarrow feature\_descriptor(x')$ ▷ Calculate and record the behavioral and structural similarity between x' and the original malware
      $e' \leftarrow evasiveness(x')$ ▷ Record the evasiveness e' of x'
      if $\mathscr{E}(b') = \phi$ or $\mathscr{E}(b') > e'$ **then** ▷ If the appropriate cell is empty or its occupants's evasiveness is $>=$ e', then
        $\mathscr{E}(b') \leftarrow e'$ ▷ store the value for evasiveness of x' in the map of elites according to its feature descriptor b'
        $\mathscr{X}(b') \leftarrow x'$ ▷ store the solution x' in the map of elites according to its feature descriptor b'
      end if
    else
      delete x'
    end if
  end for
  return feature-evasiveness map ($\mathscr{E}$ and $\mathscr{X}$)
end procedure

K. O. Babaagba, **Z. Tan**, and E. Hart, "Automatic Generation of Adversarial Metamorphic Malware Using MAP-Elites," in 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation, P.A. Castillo et al, Ed. Seville: Springer-Verlag New York, Inc., 2020, pp. 117-132.

MAL_EA VS MAP-Elites - Experimental Setting

- Experimental Settings

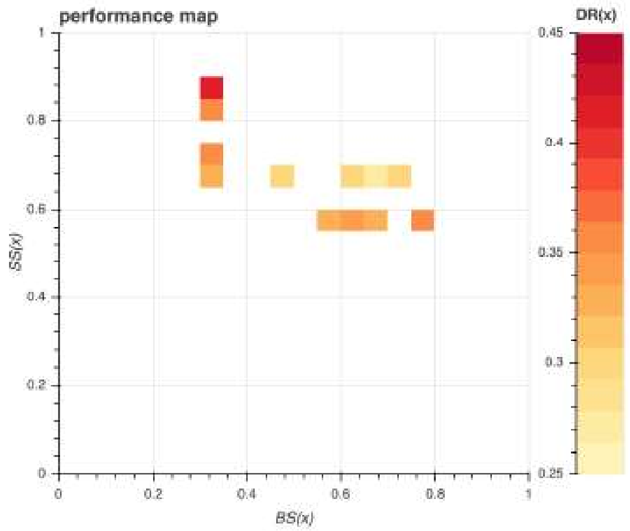
MAL_EA	
Parameter	**Setting**
Selection	Tournament
Population size	20
Iterations	120
Mutation rate	1

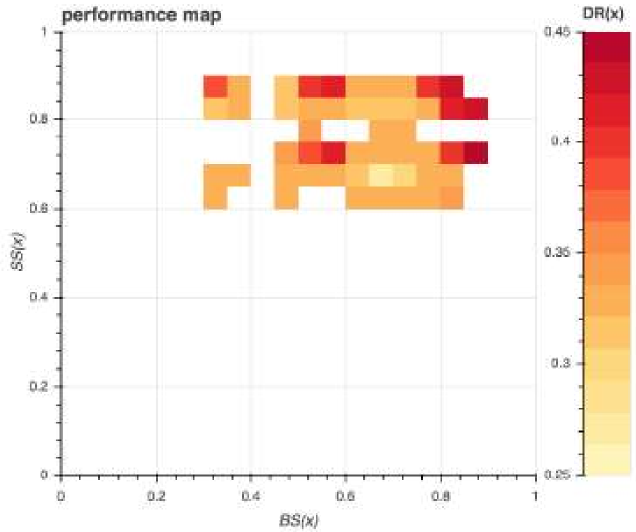
MAP-Elites	
Parameter	**Setting**
Selection	Random Selection
Bootstrap	20
Iterations	120
Mutation rate	1

K. O. Babaagba, **Z. Tan**, and E. Hart, "Automatic Generation of Adversarial Metamorphic Malware Using MAP-Elites," in 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation, P.A. Castillo et al, Ed. Seville: Springer-Verlag New York, Inc., 2020, pp. 117-132.

MAL_EA VS MAP-Elites - Results

- Performance map of MAL_EA and MAP-Elites for Dougalek family

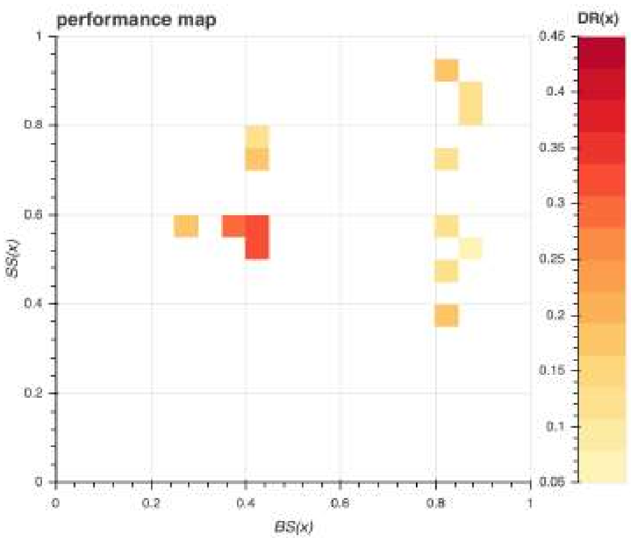


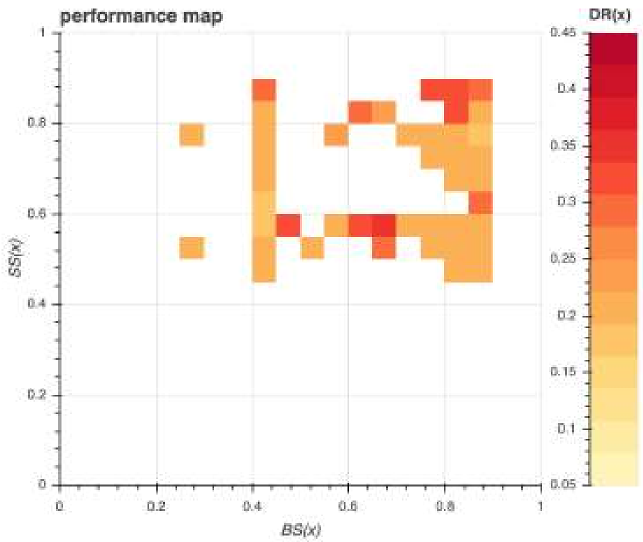


K. O. Babaagba, **Z. Tan**, and E. Hart, "Automatic Generation of Adversarial Metamorphic Malware Using MAP-Elites," in 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation, P.A. Castillo et al, Ed. Seville: Springer-Verlag New York, Inc., 2020, pp. 117-132.

MAL_EA VS MAP-Elites - Results

- Performance map of MAL_EA and MAP-Elites for Droidkungfu family

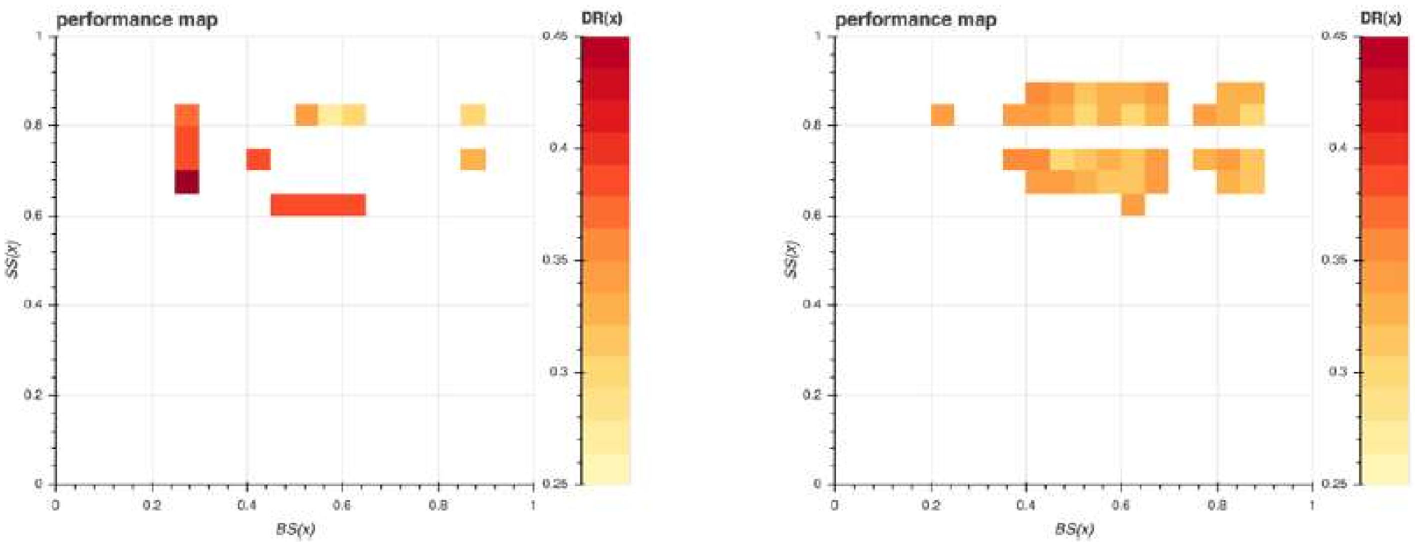




K. O. Babaagba, **Z. Tan**, and E. Hart, "Automatic Generation of Adversarial Metamorphic Malware Using MAP-Elites," in 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation, P.A. Castillo et al, Ed. Seville: Springer-Verlag New York, Inc., 2020, pp. 117-132.

MAL_EA VS MAP-Elites - Results

- Performance map of MAL_EA and MAP-Elites for GGtracker family



K. O. Babaagba, **Z. Tan**, and E. Hart, "Automatic Generation of Adversarial Metamorphic Malware Using MAP-Elites," in 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation, P.A. Castillo et al, Ed. Seville: Springer-Verlag New York, Inc., 2020, pp. 117-132.

MAL_EA VS MAP-Elites - Results

- Quantitative Comparison of MAP-Elites and MAL_EA

		Performance	Coverage	Reliability	Precision
Dougalek	MAP-Elites	**0.94**	**0.5**	**0.48**	0.96
	MAL_EA	0.92	0.06	0.06	0.97
Droidkungfu	MAP-Elites	0.85	**0.49**	**0.46**	0.94
	MAL_EA	0.86	0.07	0.07	**0.97**
GGtracker	MAP-Elites	**0.86**	**0.5**	**0.47**	0.93
	MAL_EA	0.7	0.08	0.08	0.95

K. O. Babaagba, **Z. Tan**, and E. Hart, "Automatic Generation of Adversarial Metamorphic Malware Using MAP-Elites," in 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation, P.A. Castillo et al, Ed. Seville: Springer-Verlag New York, Inc., 2020, pp. 117-132.

Can Classification of Metamorphic Malware be Improved?

- Datasets
  - *B* - benign samples
  - *Mw* - malicious samples from web
  - *EM* - evolved malware
  - *EMu* - evolved malware (unseen set for test)
- Machine learning algorithms
  - Logistic Regression
  - Support Vector Machine
  - Naïve Bayes
  - Decision Trees
  - K-Nearest Neighbour
  - LSTM

K. O. Babaagba, **Z. Tan**, and E. Hart, "Nowhere metamorphic malware can hide - a biological evolution inspired detection scheme," in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Can Classification of Metamorphic Malware be Improved?

- Results -1
  - 10-fold cross validation with model that uses samples of *B* and *Mw* as training data

Model	10-fold Mean Accuracy (std)	Validation Accuracy
LR	0.889881 (0.115217)	0.9
KNN	0.910317 (0.103585)	0.95
CART	0.912103 (0.080542)	1
NB	0.937103 (0.084450)	0.95
SVM	0.899603 (0.093919)	1

  - 10-fold cross validation with model that uses samples of *B* and *EM* as training data

Model	10-fold Mean Accuracy (std)	Validation Accuracy
LR	0.903770 (0.102321)	0.9
KNN	0.871032 (0.124375)	0.85
CART	0.922817 (0.084685)	0.95
NB	0.949603 (0.062133)	0.95
SVM	0.899206 (0.071488)	075

K. O. Babaagba, **Z. Tan**, and E. Hart, "Nowhere metamorphic malware can hide - a biological evolution inspired detection scheme," in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Can Classification of Metamorphic Malware be Improved?

- Results-2
  - 10-fold cross validation with model that uses samples of B, Mw and EM as training data

Model	10-fold Mean Accuracy (std)	Validation Accuracy
LR	0.914881 (0.093706)	0.9
KNN	0.897817 (0.099469)	0.9
CART	0.913492 (0.097603)	0.9
NB	0.949603 (0.062133)	0.95
SVM	0.905556 (0.101645)	0.95

K. O. Babaagba, **Z. Tan**, and E. Hart, "Nowhere metamorphic malware can hide - a biological evolution inspired detection scheme," in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Can Classification of Metamorphic Malware be Improved?

- Results-3
  - Comparison of accuracy obtained on the unseen test set *EMu* using a Naïve Bayes model trained on 3 different training sets

Training Data	EM_u (Accuracy)
B and M_w	0.4
B and EM	0.84
B, M_w and EM	0.82

  - Comparison of accuracy obtained on the unseen test set *EMu* using an LSTM model trained on 3 different training sets

Training Data	EM_u (Accuracy)
B and M_w	0.53
B and EM	0.9
B, M_w and EM	0.62

K. O. Babaagba, **Z. Tan**, and E. Hart, "Nowhere metamorphic malware can hide - a biological evolution inspired detection scheme," in Dependability in Sensor, Cloud, and Big Data Systems and Applications, G.Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds. Singapore: Springer Singapore, 2019, pp. 369–382.

Challenges to Securing Mobile Endpoint

- Problems with traditional authentication
  - Using a Password Challenge/Face Recognition/Fingerprint
  - Access is granted if the Input Password/Face/Fingerprint is valid – <u>One time Authentication</u>
  - The device cannot detect intruders after the authentication step is performed successfully

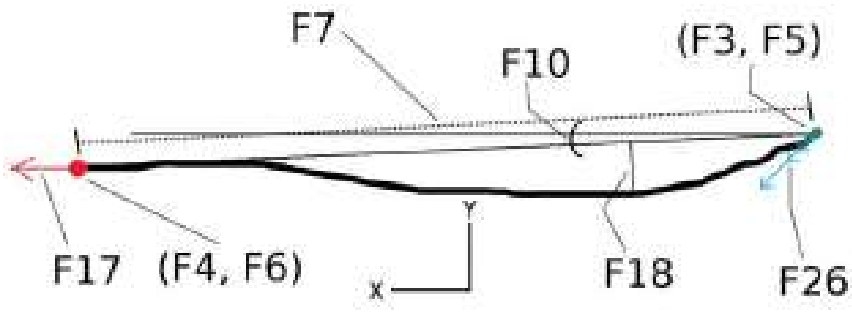
Behavioural Biometric for Continuous Authentication

Edinburgh Napier UNIVERSITY

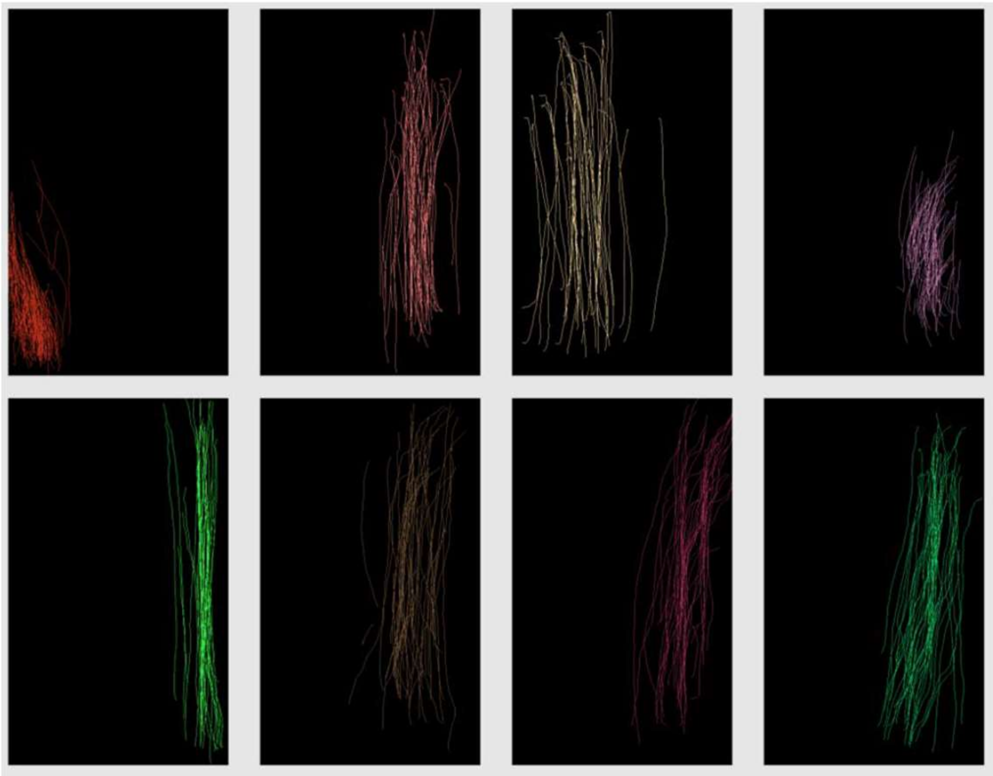
- A Behavioural Biometric driven solution
  - To authenticate users while they perform basic navigation steps on a touchscreen device
  - Without any dedicated and explicit security action that requires attention from the user

Behavioural Biometric for Continuous Authentication

- The main hypothesis
  - continuously recorded touch data from a touchscreen is distinctive enough to serve as a behavioural biometric.



Behavioural Biometric for Continuous Authentication



Behavioural Biometric for Continuous Authentication

- Two actions trigger the system logs the fingertip data
  - Sliding horizontally over the screen. Usually, one does this to browse through images or to navigate to the next page of icons in the main screen.
  - Sliding vertically over the screen to move screen content up or down. This is typically done for reading e-mail, documents or web-pages, or for browsing menus.

Behavioural Biometric for Continuous Authentication

- Continuous Authentication
  - Once the classifiers are trained, the device begins the authentication phase.
  - During this phase, the system continuously tracks all strokes and the classifier estimates if they were made by the legitimate user.
  - For consecutive negative classification results, the system resorts back to the initial entry-point based authentication method and challenges the user.

Behavioural Biometric for Continuous Authentication

- Open questions:
  - Multi-class classification VS bi-class classification VS Single-class classification
  - How to protect user privacy?
    - Federated learning
    - Homomorphic encryption
  - How to transfer models between user devices?
    - Transfer learning

References

- Aydogan, E., & Sen, S. (2015). Automatic Generation of Mobile Malwares Using Genetic Programming. *Applications of Evolutionary Computation*, 745--756. https://doi.org/10.1007/978-3-319-16549-3_60
- Baysa, D., Low, R. M., & Stamp, M. (2013). Structural entropy and metamorphic malware. *Journal in Computer Virology*, *9*(4), 179–192. https://doi.org/10.1007/s11416-013-0185-4
- Bruschi, D., Martignoni, L., & Monga, M. (2007). Code normalization for self-mutating malware. *IEEE Security and Privacy*, *5*(2), 46–54. https://doi.org/10.1109/MSP.2007.31
- Cody-Kenny, B., Lopez, E. G., & Barrett, S. (2015). locoGP: Improving Performance by Genetic Programming Java Source Code. In *Genetic Improvement 2015 Workshop* (pp. 811–818). https://doi.org/doi:10.1145/2739482.2768419
- Langdon, W. B., & Harman, M. (2015). Optimizing existing software with genetic programming. *IEEE Transactions on Evolutionary Computation*, *19*(1), 118–135. https://doi.org/10.1109/TEVC.2013.2281544
- Poli, R., Langdon, W. B., & McPhee, N. F. (2008). *A Field Guide to Genetic Programing*. *Wyvern*. Retrieved from http://www.essex.ac.uk/wyvern/2008-04/Wyvern April 08 7126.pdf
- Vigna, G., Robertson, W., & Balzarotti, D. (2004). Testing network-based intrusion detection signatures using mutant exploits. *Proceedings of the 11th ACM Conference on Computer and Communications Security*, 21–30. https://doi.org/10.1145/1030083.1030088
- White, D. R., Arcuri, A., & Clark, J. A. (2011). Evolutionary improvement of programs. *IEEE Transactions on Evolutionary Computation*, *15*(4), 515–538. https://doi.org/10.1109/TEVC.2010.2083669
- Xu, W., Qi, Y., & Evans, D. (2016). Automatically Evading Classifiers - A Case Study on PDF Malware Classifier. *Ndss*, *2016*(February).
- Vidas, T.: Contagio Mobile Malware Mini Dump (2015), http://contagiominidump.blogspot.com/2015/01/android-hideicon-malware-samples.html
- Zhou, Y., Jiang, X.: Dissecting Android malware: Characterization and evolution. In: Proceedings - IEEE Symposium on Security and Privacy (2012).
- M. Frank, R. Biedert, E. Ma, I. Martinovic and D. Song, "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication," in IEEE Transactions on Information Forensics and Security, vol. 8, no. 1, pp. 136-148, Jan. 2013.
- P. Aaby, M. Valerio Giuffrida, W. J. Buchanan and Z. Tan, "Towards Continuous User Authentication Using Personalised Touch-Based Behaviour," 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 2020, pp. 41-48, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00023.

Thank You for Listening!

Q & A

- Dr Zhiyuan Tan (Associate Professor)
- Deputy to the Research Degree Programmes Leader, School of Computing @ Edinburgh Napier University
- Homepage - https://www.napier.ac.uk/people/thomas-tan
- Email – z.tan@napier.ac.uk

- Academic Editor, Security and Communication Networks, ISSN: 1939-0114 (Print) ISSN: 1939-0122 (Online) DOI: 10.1155/2037
- Section Editor, Ad Hoc & Sensor Wireless Networks, ISSN: 1551-9899
- Topics Board Editor, Network, ISSN: 2673-8732