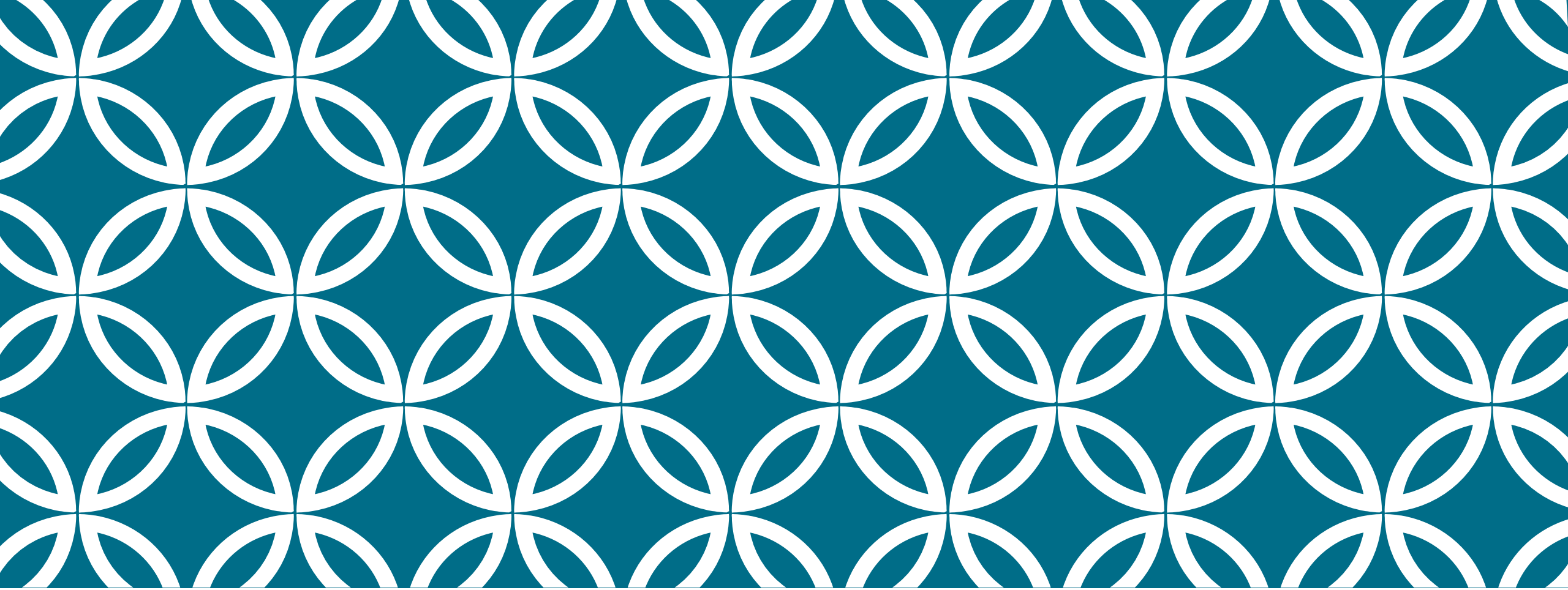


"I DON'T KNOW TOO MUCH ABOUT IT": ON THE SECURITY MINDSETS OF COMPUTER SCIENCE STUDENTS

Mohammad Tahaei
mohammad.tahaei@ed.ac.uk
Adam Jenkins, Kami Vaniea, Maria Wolters

University of Edinburgh
September 2019

- Software is everywhere.
- Security expectations from the public.
- Yet software security is an issue.
- Developers much like other users need support!
- Research area: Developer-Centred Security (DCS).

MOTIVATION

MOTIVATION

- Software developers come from various backgrounds. Several of them have formal education in CS.
- 63.3% of professional developers majored in CS-related fields [Stack Overflow, 2019, 66823 responses].
- Yet, we know little about students mindsets around computer security.
- One potential opportunity for changing developers' security attitudes and practices is during their training.

RESEARCH QUESTIONS

1. What are students' comprehension of S&P related concepts?
2. To what extent do students consider S&P while coding applications, and how do they implement it?

METHOD

Method

- Semi-structured interviews
- ~68 minutes long

Advertisement:

- Mailing lists, Facebook …

Participants:

- 6 BSc, 11 MSc, and 3 PhD students
- Half took a security course

Analysis:

- Qualitative coding and thematic analysis

Question topics

App design

Threats/Hackers

Coding experiences

(Security) APIs

Software engineering

FREE LISTING ACTIVITY: WHAT DOES 'COMPUTER SECURITY' MEAN?

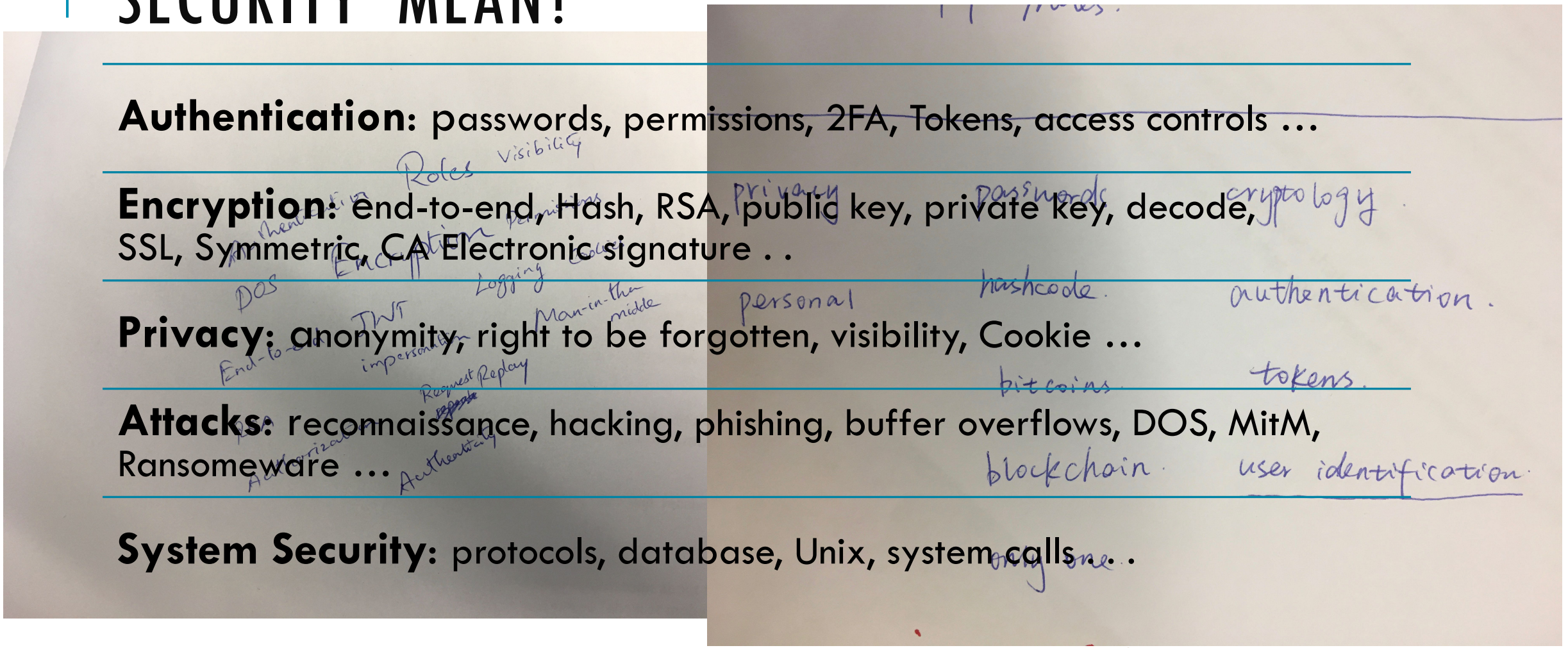
Authentication: passwords, permissions, 2FA, Tokens, access controls …

Encryption: end-to-end, Hash, RSA, public key, private key, decode, SSL, Symmetric, CA Electronic signature . .

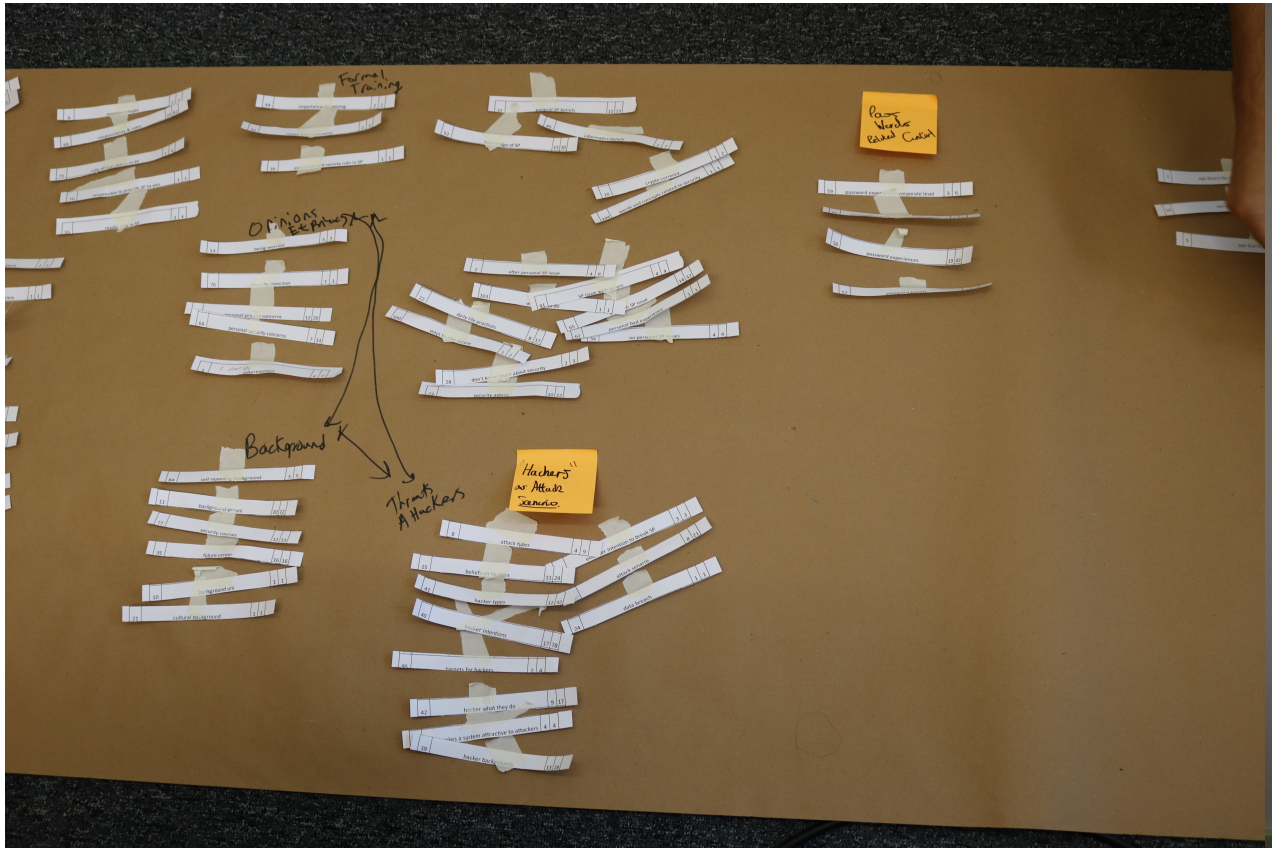
Privacy: anonymity, right to be forgotten, visibility, Cookie …

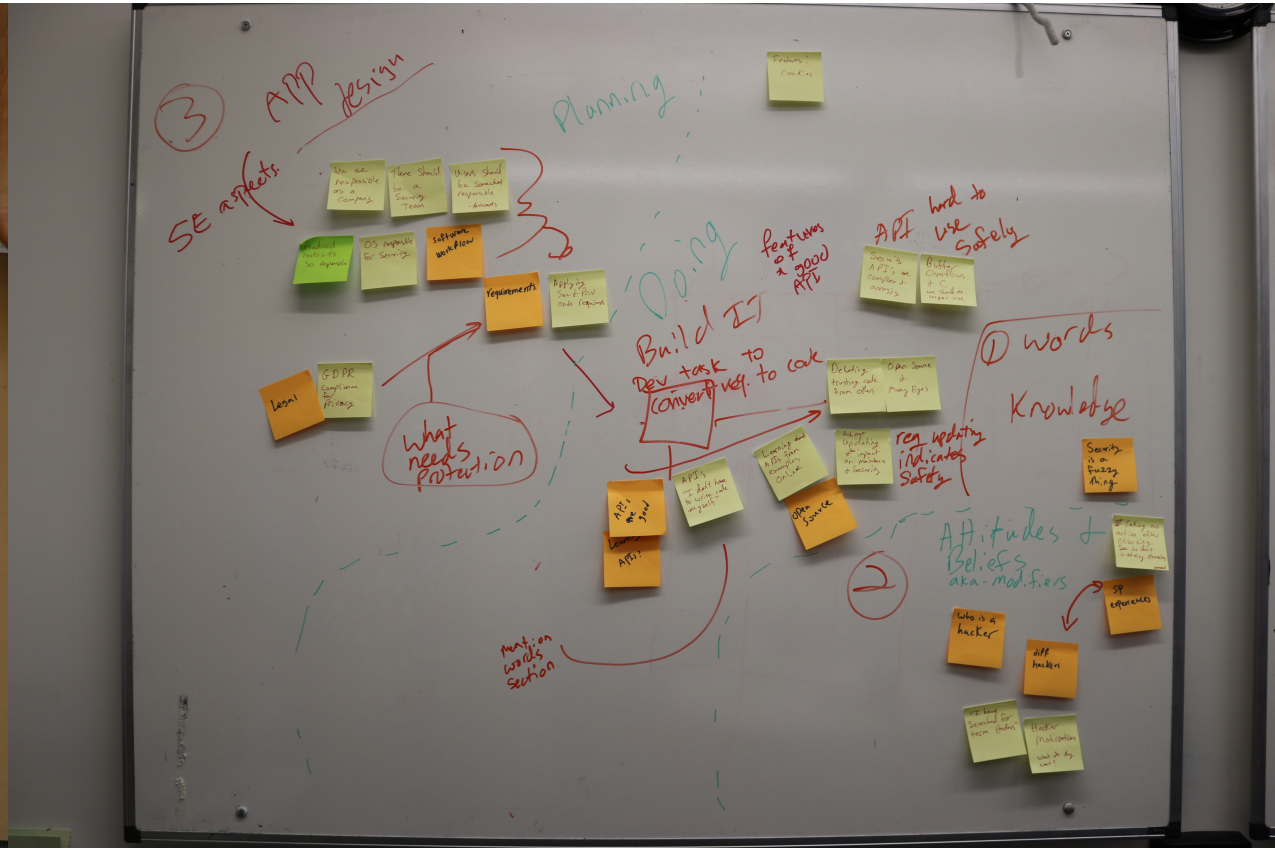
Attacks: reconnaissance, hacking, phishing, buffer overflows, DOS, MitM, Ransomeware …

System Security: protocols, database, Unix, system calls . . .



ANALYSIS: THEMATIC ANALYSIS/ AFFINITY DIAGRAM





WHO ARE HACKERS AND WHAT DO THEY WANT?

Observed similar threat models to Wash, folk models of home computer security

- Graffiti: attacker with high technical background
- Burglar: those who commit crimes using computers mostly with financial motivations
- Big fish: hackers looking for high valued targets
- Contractor: graffiti hackers with financial/criminal motivations

WHO ARE HACKERS AND WHAT DO THEY WANT?

Big fish: “Political incentive that certain countries fund a lot of hacking and cracking to gain power depending how important or how famous you are there might be people who want to get access to your account.” [PS13]

Contractor: “Trained people who are trained to do this kind of stuff. Either by some governments to hack other governments. Or to break the encryption or security mechanism.” [P02]

REQUIREMENTS

Requirements & responsibilities: playing hot potato

- Security team?
- Implied feature?
- When asked to design an in-class discussion app, only 4 mentioned security or privacy elements.

> "There should be a security team. Which takes care of that. Just like any other team inside the company. Like UI, testing team." [PS04]

TOOLS

APIs, building on other peoples' code

- Useful and handy.
- Security APIs?
- Open source: trusting other peoples' code.

TOOLS: PATTERNS SIMILAR TO PRIOR WORK

"Sometimes just some posts either forums or some question and answer community like **Stack Overflow**. There are people show you how to use in their answers, kind of you can **copy paste** and modify that to suit your needs." [P05]

"If I look at the code base and see something on Github and it has let's say 2000 stars. Few hundred people watching it. The **code is all open**. I tend to perhaps foolishly I assume that if this **many people have looked at it** and if there was something up. **Surely someone would do have said something.**" [P08]

CONCLUSIONS & FUTURE WORK

- CS students have a range of hacker mindsets, lack of experience with security APIs, a mixed view of who is in charge of S&P in the software life cycle, and a tendency to trust other peoples' code as a convenient approach to rapidly build software.
- Attitudes of students match many of those observed by other researchers looking at professional level developers.
- Comparing industry and professional developers with students.
- Impact of open source and code reuse in system security. Trust in others' code.

Motivation	Security attitudes and approaches of software developers have a large impact on the software.
	Yet, we know very little about how and when these views are constructed.
Method	Semi-structured interviews with computer science students.
	Qualitative coding & affinity diagrams.
Results	A range of hacker and attack mindsets.
	Lack of experience with security APIs.
	A mixed view of who is in charge of S&P in the software life cycle.
	A tendency to trust other peoples' code.

THE UNIVERSITY *of* EDINBURGH
informatics

Mohammad Tahaei

mohammad.tahaei@ed.ac.uk

Adam Jenkins

Kami Vaniea

Maria Wolters

"I DON'T KNOW TOO MUCH ABOUT IT": ON THE SECURITY MINDSETS OF COMPUTER SCIENCE STUDENTS

Table 1. Interview study demographics. P = participant without computer security background; PS = participant who self-describes as having taken a computer security course in the past.

Participant	Gender	Nationality	Age	Expected Degree
PS01	M	EU	29	PhD
P02	M	EU	28	MSc
PS03	F	Asia	22	MSc
PS04	M	Asia	24	MSc
PS05	M	Asia	25	PhD
P06	F	Asia	23	MSc
P07	M	Asia	22	BSc
PS08	M	UK	21	MSc
PS09	M	Asia	25	MSc
P10	M	Asia	21	BSc
P11	M	EU	22	BSc
PS12	M	Asia	23	MSc
PS13	M	EU	21	BSc
P14	M	EU	20	BSc
PS15	M	EU	25	PhD
PS16	M	Asia	37	MSc
P17	F	EU	25	BSc
P18	F	Asia	23	MSc
P19	M	UK	24	MSc
P20	F	Asia	20	MSc

Table 2. Topics mentioned during free-listing, number of words participants listed associated with that topic, number of unique participants listing at least one word associated with the topic, and a set of sample words representing the range.

Topic	#Words	#Participants	Example words
Encryption	28	11	End-to-end, hash, RSA, public/private key, SSL, symmetric.
Authentication	28	9	Passwords, permissions, 2FA, tokens, access controls, emails.
Privacy	27	10	Anonymity, right to be forgotten, visibility, cookies.
Attacks	25	8	Reconnaissance, phishing, buffer overflows, DoS, MITM.
System security	13	5	Protocols, database, Unix, system calls, TCP/IPs.
Social	13	7	Regulations, roles, responsibilities, public knowledge.
Finance	8	4	PayPal, Apple Pay, Bitcoin, online payments.
Defending	7	5	Anti-virus/malware, penetration testing, logging, bounties.
Security holes	5	4	Failures, physical access, loopholes.
Companies	5	3	Facebook, Google, Norton, Red Hat.
Trade offs	4	3	Usable security, features vs security, easy to use UX.