# Introducing Usability to the Common Criteria

Luke Church  Matthew Nicolas Kreeger  Marcus Streets

University of Cambridge, Computer Laboratory
Cambridge, England, U.K.
Email: luke@church.name

nCipher Corporation
Cambridge, England, U.K.
E-mail: {matthew, marcus}@ncipher.com

9th International Common Criteria Conference (ICCC), Jeju, Korea, 2008

# Talk Synopsis

- Common Criteria evaluations need to include usability
- Security and usability are entangled
- Show where usability failings have resulted in security failures
- Conclude with the presentation of a new CC class

# Talk Synopsis

- Common Criteria evaluations need to include usability
- Security and usability are entangled
- Show where usability failings have resulted in security failures
- Conclude with the presentation of a new CC class

# Talk Synopsis

- Common Criteria evaluations need to include usability
- Security and usability are entangled
- Show where usability failings have resulted in security failures
- Conclude with the presentation of a new CC class

# Talk Synopsis

- Common Criteria evaluations need to include usability
- Security and usability are entangled
- Show where usability failings have resulted in security failures
- Conclude with the presentation of a new CC class

# Talk Synopsis

- Common Criteria evaluations need to include usability
- Security and usability are entangled
- Show where usability failings have resulted in security failures
- Conclude with the presentation of a new CC class

# Introduction

- "Usability is a problem"
  - Anderson: real world failures
  - Cranor: phishing
  - Schneier: Microsoft Vista's UAC
  - Whitten: analysis of PGP
- Usability is dominating many technical discussions regarding security

# Introduction

- "Usability is a problem"
  - Anderson: real world failures
  - Cranor: phishing
  - Schneier: Microsoft Vista's UAC
  - Whitten: analysis of PGP
- Usability is dominating many technical discussions regarding security

# Introduction

- “Usability is a problem”
  - Anderson: real world failures
  - Cranor: phishing
  - Schneier: Microsoft Vista’s UAC
  - Whitten: analysis of PGP
- Usability is dominating many technical discussions regarding security

# Introduction

- “Usability is a problem”
  - Anderson: real world failures
  - Cranor: phishing
  - Schneier: Microsoft Vista’s UAC
  - Whitten: analysis of PGP
- Usability is dominating many technical discussions regarding security

# Introduction

- Purpose of CC to provide assurance that a system is secure
  - “in theory” is not sufficient
- PGP as an example:
  - reasonably secure “in theory”
  - 25% of the study participants accidentally revealed their secret information

# Introduction

- Purpose of CC to provide assurance that a system is secure
  - "in theory" is not sufficient
- PGP as an example:
  - reasonably secure "in theory"
  - 25% of the study participants accidentally revealed their secret information

# Introduction

- Purpose of CC to provide assurance that a system is secure
  - “in theory” is not sufficient
- PGP as an example:
  - reasonably secure “in theory”
  - 25% of the study participants accidentally revealed their secret information

# Introduction

- Purpose of CC to provide assurance that a system is secure
  - “in theory” is not sufficient
- PGP as an example:
  - reasonably secure “in theory”
  - 25% of the study participants accidentally revealed their secret information

# Introduction

- Purpose of CC to provide assurance that a system is secure
  - “in theory” is not sufficient
- PGP as an example:
  - reasonably secure “in theory”
  - 25% of the study participants accidentally revealed their secret information

# Introduction

- In providing product assurance:
  - technical evaluation alone is insufficient
  - assessing product behaviour in the real world is necessary
- Evaluating the security of a system without considering the interaction between itself and the users leads CC to provide less assurance than it might

# Introduction

- In providing product assurance:
  - technical evaluation alone is insufficient
  - assessing product behaviour in the real world is necessary
- Evaluating the security of a system without considering the interaction between itself and the users leads CC to provide less assurance than it might

# Introduction

- In providing product assurance:
  - technical evaluation alone is insufficient
  - assessing product behaviour in the real world is necessary
- Evaluating the security of a system without considering the interaction between itself and the users leads CC to provide less assurance than it might

# Introduction

- In providing product assurance:
  - technical evaluation alone is insufficient
  - assessing product behaviour in the real world is necessary
- Evaluating the security of a system without considering the interaction between itself and the users leads CC to provide less assurance than it might

# Pragmatics

- Systems with unusable security get bypassed:
  - Regular warnings lead to muscle memory responses
  - If security is seen as “getting in the way” it is ignored
  - The security of a system may not be as strong as it appears
- Human errors must be accounted for:
  - Range of heuristics and biases which affect human decision making e.g. confirmation bias
  - “post completion error”
  - The way individuals interact with a system has a substantial effect on its security properties

# Pragmatics

- Systems with unusable security get bypassed:
  - Regular warnings lead to muscle memory responses
  - If security is seen as “getting in the way” it is ignored
  - The security of a system may not be as strong as it appears
- Human errors must be accounted for:
  - Range of heuristics and biases which affect human decision making e.g. confirmation bias
  - “post completion error”
  - The way individuals interact with a system has a substantial effect on its security properties

# Pragmatics

- Systems with unusable security get bypassed:
  - Regular warnings lead to muscle memory responses
  - If security is seen as “getting in the way” it is ignored
  - The security of a system may not be as strong as it appears
- Human errors must be accounted for:
  - Range of heuristics and biases which affect human decision making e.g. confirmation bias
  - “post completion error”
  - The way individuals interact with a system has a substantial effect on its security properties

# Pragmatics

- Systems with unusable security get bypassed:
  - Regular warnings lead to muscle memory responses
  - If security is seen as “getting in the way” it is ignored
  - The security of a system may not be as strong as it appears
- Human errors must be accounted for:
  - Range of heuristics and biases which affect human decision making e.g. confirmation bias
  - “post completion error”
  - The way individuals interact with a system has a substantial effect on its security properties

# Pragmatics

- Systems with unusable security get bypassed:
  - Regular warnings lead to muscle memory responses
  - If security is seen as “getting in the way” it is ignored
  - The security of a system may not be as strong as it appears
- Human errors must be accounted for:
  - Range of heuristics and biases which affect human decision making e.g. confirmation bias
  - “post completion error”
  - The way individuals interact with a system has a substantial effect on its security properties

# Social interactions

- The social context in which an application is used effects the behaviour of the users
- Results in a difference between the theoretical and practical security properties of the system
- Commonly regarded that the relationship between technology and its social use is complex
- Failing to account for social context can result in theoretical security differing from actual security

# Social interactions

- The social context in which an application is used effects the behaviour of the users
- Results in a difference between the theoretical and practical security properties of the system
- Commonly regarded that the relationship between technology and its social use is complex
- Failing to account for social context can result in theoretical security differing from actual security

# Social interactions

- The social context in which an application is used effects the behaviour of the users
- Results in a difference between the theoretical and practical security properties of the system
- Commonly regarded that the relationship between technology and its social use is complex
- Failing to account for social context can result in theoretical security differing from actual security

# Social interactions

- The social context in which an application is used effects the behaviour of the users
- Results in a difference between the theoretical and practical security properties of the system
- Commonly regarded that the relationship between technology and its social use is complex
- Failing to account for social context can result in theoretical security differing from actual security

# Social interactions

- The social context in which an application is used effects the behaviour of the users
- Results in a difference between the theoretical and practical security properties of the system
- Commonly regarded that the relationship between technology and its social use is complex
- Failing to account for social context can result in theoretical security differing from actual security

# Systemic issues

- The dependency of security on usability is not accidental
  - It is a fundamental part of security engineering
- CC explicitly acknowledges that there may be non-technical controls (A.6.3)
  - aims to provide confidence in the technical component
  - technical component interacts with other components to be considered useful
  - CC acknowledges this with interface assurance (ADV_FSP)

# Systemic issues

- The dependency of security on usability is not accidental
  - It is a fundamental part of security engineering
- CC explicitly acknowledges that there may be non-technical controls (A.6.3)
  - aims to provide confidence in the technical component
  - technical component interacts with other components to be considered useful
  - CC acknowledges this with interface assurance (ADV_FSP)

# Systemic issues

- The dependency of security on usability is not accidental
  - It is a fundamental part of security engineering
- CC explicitly acknowledges that there may be non-technical controls (A.6.3)
  - aims to provide confidence in the technical component
  - technical component interacts with other components to be considered useful
  - CC acknowledges this with interface assurance (ADV_FSP)

# Systemic issues

- The dependency of security on usability is not accidental
  - It is a fundamental part of security engineering
- CC explicitly acknowledges that there may be non-technical controls (A.6.3)
  - aims to provide confidence in the technical component
  - technical component interacts with other components to be considered useful
  - CC acknowledges this with interface assurance (ADV_FSP)

# Systemic issues

- The dependency of security on usability is not accidental
  - It is a fundamental part of security engineering
- CC explicitly acknowledges that there may be non-technical controls (A.6.3)
  - aims to provide confidence in the technical component
  - technical component interacts with other components to be considered useful
  - CC acknowledges this with interface assurance (ADV_FSP)

# Systemic issues

- Ignoring usability is to ignore the interaction between the human and technical components of a system
- No complete assurance of the security of the product can be given
- Essential to consider this interaction in terms of the social context
- Cannot assure security without assuring usability

# Systemic issues

- Ignoring usability is to ignore the interaction between the human and technical components of a system
- No complete assurance of the security of the product can be given
- Essential to consider this interaction in terms of the social context
- Cannot assure security without assuring usability

# Systemic issues

- Ignoring usability is to ignore the interaction between the human and technical components of a system
- No complete assurance of the security of the product can be given
- Essential to consider this interaction in terms of the social context
- Cannot assure security without assuring usability

# Systemic issues

- Ignoring usability is to ignore the interaction between the human and technical components of a system
- No complete assurance of the security of the product can be given
- Essential to consider this interaction in terms of the social context
- Cannot assure security without assuring usability

# Systemic issues

- Ignoring usability is to ignore the interaction between the human and technical components of a system
- No complete assurance of the security of the product can be given
- Essential to consider this interaction in terms of the social context
- Cannot assure security without assuring usability

# Problems of not evaluating usability within the CC?

- Prevents the CC from making general claims about security
- Restricts security to being a specified set of technical properties
- Issues arising from not evaluating usability within the CC:
  - False perception as to what is assured
  - Encourages unrealistic expectations of a user's capabilities
  - Provides an incentive to migrate security decisions to the user
  - Fails to motivate progress in security usability

# Problems of not evaluating usability within the CC?

- Prevents the CC from making general claims about security
- Restricts security to being a specified set of technical properties
- Issues arising from not evaluating usability within the CC:
  - False perception as to what is assured
  - Encourages unrealistic expectations of a user's capabilities
  - Provides an incentive to migrate security decisions to the user
  - Fails to motivate progress in security usability

# Problems of not evaluating usability within the CC?

- Prevents the CC from making general claims about security
- Restricts security to being a specified set of technical properties
- Issues arising from not evaluating usability within the CC:
  - False perception as to what is assured
  - Encourages unrealistic expectations of a user's capabilities
  - Provides an incentive to migrate security decisions to the user
  - Fails to motivate progress in security usability

# Problems of not evaluating usability within the CC?

- Prevents the CC from making general claims about security
- Restricts security to being a specified set of technical properties
- Issues arising from not evaluating usability within the CC:
  - False perception as to what is assured
  - Encourages unrealistic expectations of a user's capabilities
  - Provides an incentive to migrate security decisions to the user
  - Fails to motivate progress in security usability

# Problems of not evaluating usability within the CC?

- Prevents the CC from making general claims about security
- Restricts security to being a specified set of technical properties
- Issues arising from not evaluating usability within the CC:
  - False perception as to what is assured
  - Encourages unrealistic expectations of a user's capabilities
  - Provides an incentive to migrate security decisions to the user
  - Fails to motivate progress in security usability

# Including usability within the CC

- We propose adding a new class to the CC to cover user identification
- Dependency between this class and others to ensure information carried through to the rest of the design process

# Including usability within the CC

- We propose adding a new class to the CC to cover user identification
- Dependency between this class and others to ensure information carried through to the rest of the design process

# Including usability within the CC

- We propose adding a new class to the CC to cover user identification
- Dependency between this class and others to ensure information carried through to the rest of the design process

# Class AUI: User identification

**User Identification (AUI_USR)**

*Objectives:*
A common failing in usability is failing to identify correctly who your users are. The current user base of computer systems varies widely, from computer scientists to home users and children. It is crucial that the product is designed with its users in mind. Companies often design for the wrong user group. It is essential that this does not occur. Consequently effort needs to have been invested to determine that the believed user base is the correct one.

# Class AUI: User identification

**AUI_USR.1**

*Dependencies:* No dependencies.

*Developer action element:*

**AUI_USR.1.1D** The developer shall provide a statement that identifies the class of person that is expected to use the TOE.

*Content and presentation elements:*

**AUI_USR.1.1C** The statement shall include information on the users expected educational attainments and areas of technical competence.

# Class AUI: User identification

**AUI_USR.2**

*Dependencies:* AUI_USR.1

*Developer action element:*

**AUI_USR.2.1D** The developer shall provide evidence that supports the claims in AUI_USR.1.

*Content and presentation elements:*

**AUI_USR.2.1C** The evidence shall demonstrate that the vendor has taken best efforts to identify the users for the TOE.

# Class AUI: User identification

**Social Context (AUI_SCT)**

*Objectives:*
The social context in which security technology is used has a substantial effect on the behaviour of the users.
Failing to account for the social context in which technology is used can result in the apparent security properties of the system differing very substantially from actual security properties of the system in use.

# Class AUI: User identification

**AUI_SCT.1**

*Dependencies:* AUI_USR.1

*Developer action element:*

**AUI_SCT.1.1D** The developer shall provide a statement that identifies the social context in which the TOE is expected to be used.

# Class AUI: User identification

**AUI_SCT.2**

*Dependencies:* AUI_SCT.1

*Developer action element:*

**AUI_SCT.2.1D** The developer shall provide evidence that supports the claims in AUI_SCT.1.

*Content and presentation elements:*

**AUI_SCT.2.1C** The evidence shall demonstrate that the vendor has taken best efforts to identify the social context in which the TOE will be deployed.

# Class AUI: User identification

**Usability (AUI_USE)**

*Objectives:*
The objective of this family is to ensure that the security features of the TOE can be operated correctly by the identified user in the identified social context.
Given the wide range of possible TOEs that can be submitted it is not possible to enumerate the tests that could be carried out. It is the duty of the evaluator to determine whether the information provided by the vendor satisfies this requirement for the specific TOE under test.

# Class AUI: User identification

**AUI_USE.1**

*Dependencies:* AUI_USR.1, AUI_SCT.1, AGD_OPE.1, AGD_PRE.1

*Developer action element:*

**AUI_USE.1.1D** The developer shall provide evidence to demonstrate that the security principles and interfaces of the TOE as set out in the user guidance are understandable and usable by the identified users.

# Class AUI: User identification

We would suggest the elements be introduced at the following Common Criteria Assurance Levels:

- EAL 1 AUI_USR.1, AUI_USE.1
- EAL 2 add AUI_SCT.1
- EAL 3 add AUI_USR.2
- EAL 4 add AUI_SCT.2

# Concluding Remarks

- Security and usability are inherently entangled
- The introduction of usability to security focused evaluation and assurance schemes has to some extent been overlooked
- We have attempted to address this through the introduction of a user identification class

# Concluding Remarks

- Security and usability are inherently entangled
- The introduction of usability to security focused evaluation and assurance schemes has to some extent been overlooked
- We have attempted to address this through the introduction of a user identification class

# Concluding Remarks

- Security and usability are inherently entangled
- The introduction of usability to security focused evaluation and assurance schemes has to some extent been overlooked
- We have attempted to address this through the introduction of a user identification class

# Concluding Remarks

- Security and usability are inherently entangled
- The introduction of usability to security focused evaluation and assurance schemes has to some extent been overlooked
- We have attempted to address this through the introduction of a user identification class