

# Cyberpatterns 2012

Proceedings of the First International Workshop on  
Cyberpatterns: Unifying Design Patterns with  
Security, Attack and Forensic Patterns

**9-10 July 2012**

The Cosener's House, Abingdon, UK

Organized and Sponsored by

**Oxford Brookes University**

Sponsored by

**SOPHOS**

In association with

**BCS Information Security  
Specialist Group**

**BCS Formal Aspects of Computing  
Specialist Group**

**BCS Cybercrime Forensics  
Specialist Group**

# Cyberpatterns

[http://tech.brookes.ac.uk/  
CyberPatterns2012](http://tech.brookes.ac.uk/CyberPatterns2012)

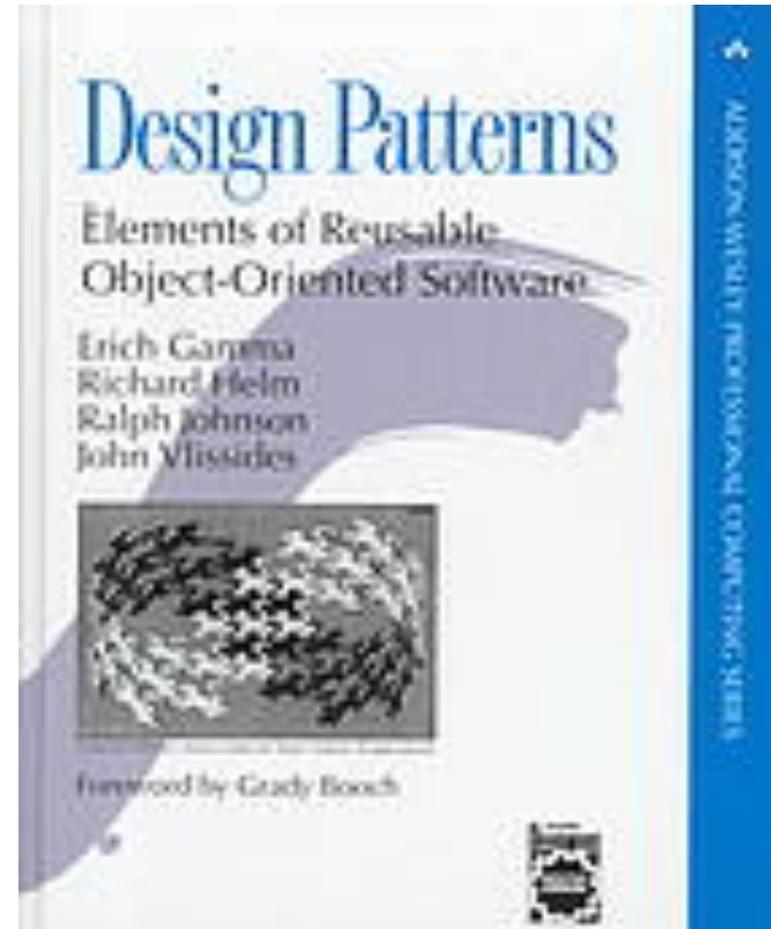
Clive Blackwell  
Computing and Communication Technologies  
Faculty of Technology, Design and Environment  
Oxford Brookes University  
Wheatley Campus, Wheatley,  
Oxford OX33 1HX  
[cblackwell@brookes.ac.uk](mailto:cblackwell@brookes.ac.uk)

# Cyberpatterns workshop

- Unifying Design Patterns with Security, Attack and Forensic Patterns
- Held at Abingdon, near Oxford (9-10 July 2012)
  - Sponsored by Oxford Brookes University and Sophos
- Keynote speakers
  - Sean Barnum (MITRE) on CAPEC and CybOX
  - Kevin Lano (King's College London) formalising design patterns and UML
- 37 participants, 19 accepted papers
- Organisations attending
  - Private – BT, Sophos, Aurora Consulting
  - Government – CESG
  - Non-profit – MITRE, JANET, Nominet
  - Universities – Abertay, Dartmouth, Glasgow, King's College London, Kingston, Lancaster, Liverpool John Moores, Newcastle, Oxford, Oxford Brookes, University College London, Warwick, West London
- Further workshops planned
  - Already promised funding, which was not even requested
  - Would like a separate industry day for Cyberpatterns
    - Bridge the gap between theory and practice

# Background

- Concept based on analogy with architectural design patterns (Christopher Alexander)
- ‘Design Patterns’ by Gamma, Helm, Vlissides, Johnson (1994), was key text that popularised software design patterns
  - ‘Gang of four’ (GoF) book
- Introduced 23 patterns, classified as Creational, Structural or Behavioural
- Software design patterns widely used for representing solutions to recurring design problems
- Subsequent identification and application of patterns in wide range of software domains, including security
- Led eventually to security and attack patterns (eg CAPEC)



# Design pattern template (GoF)

**Pattern Name and Classification:** A descriptive and unique name that helps in identifying and referring to the pattern.

**Intent:** A description of the goal behind the pattern and the reason for using it.

**Also Known As:** Other names for the pattern.

**Motivation (Forces):** A scenario consisting of a problem and a context in which this pattern can be used.

**Applicability:** Situations in which this pattern is usable; the context for the pattern.

**Structure:** A graphical representation of the pattern. Class diagrams and Interaction diagrams may be used for this purpose.

**Participants:** A listing of the classes and objects used in the pattern and their roles in the design.

**Collaboration:** A description of how classes and objects used in the pattern interact with each other.

**Consequences:** A description of the results, side effects, and trade offs caused by using the pattern.

**Implementation:** A description of an implementation of the pattern; the solution part of the pattern.

**Sample Code:** An illustration of how the pattern can be used in a programming language.

**Known Uses:** Examples of real usages of the pattern.

**Related Patterns:** Other patterns that have some relationship with the pattern; discussion of the differences between the pattern and similar patterns.

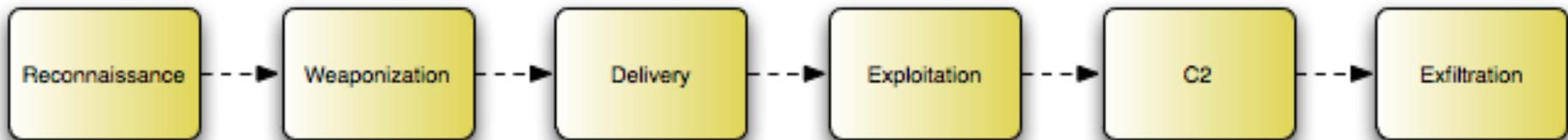
© Wikipedia 2012

# Cyberpatterns

- Initial theme: Unifying Design Patterns with Security, Attack and Forensic Patterns
- Inspiration for workshop from applying longstanding work on formalising design patterns to security
  - Hong Zhu and Ian Bayley (PATTERNS best paper, ACM TOSE etc)
- Widened scope with several papers on pattern matching
  - Need to explore the link between the two ideas of patterns
- Large quantity of existing work on patterns in security
  - Substantial catalogue of attack patterns (CAPEC)
  - Body of knowledge of security patterns (CERT/CMU SEI report)
- Patterns in digital forensics could also be significant
- Applications to cloud, critical infrastructure, cyber warfare, IoT
- Formulating an initial roadmap for theoretical and practical progress
  - Looking for opportunities from an encompassing pattern framework

# Attack patterns benefits

- Help to understand adversary motivation, intent and behaviour
  - ‘Know your enemy’ (Sun Zhu – The Art of War)
- Help provide better overview of incidents
  - Helps to understand the overall context and information that may be available to counter incidents
- Potentially better than matching attacks with strings or keywords
  - Can match parts of attack pattern template with fields that can be observed, whereas others need to be inferred
- May be able to stop incidents in more places with better understanding of context
  - Incident model demonstrates multiple places an attack can be stopped
  - Kill chain – move left of exploit

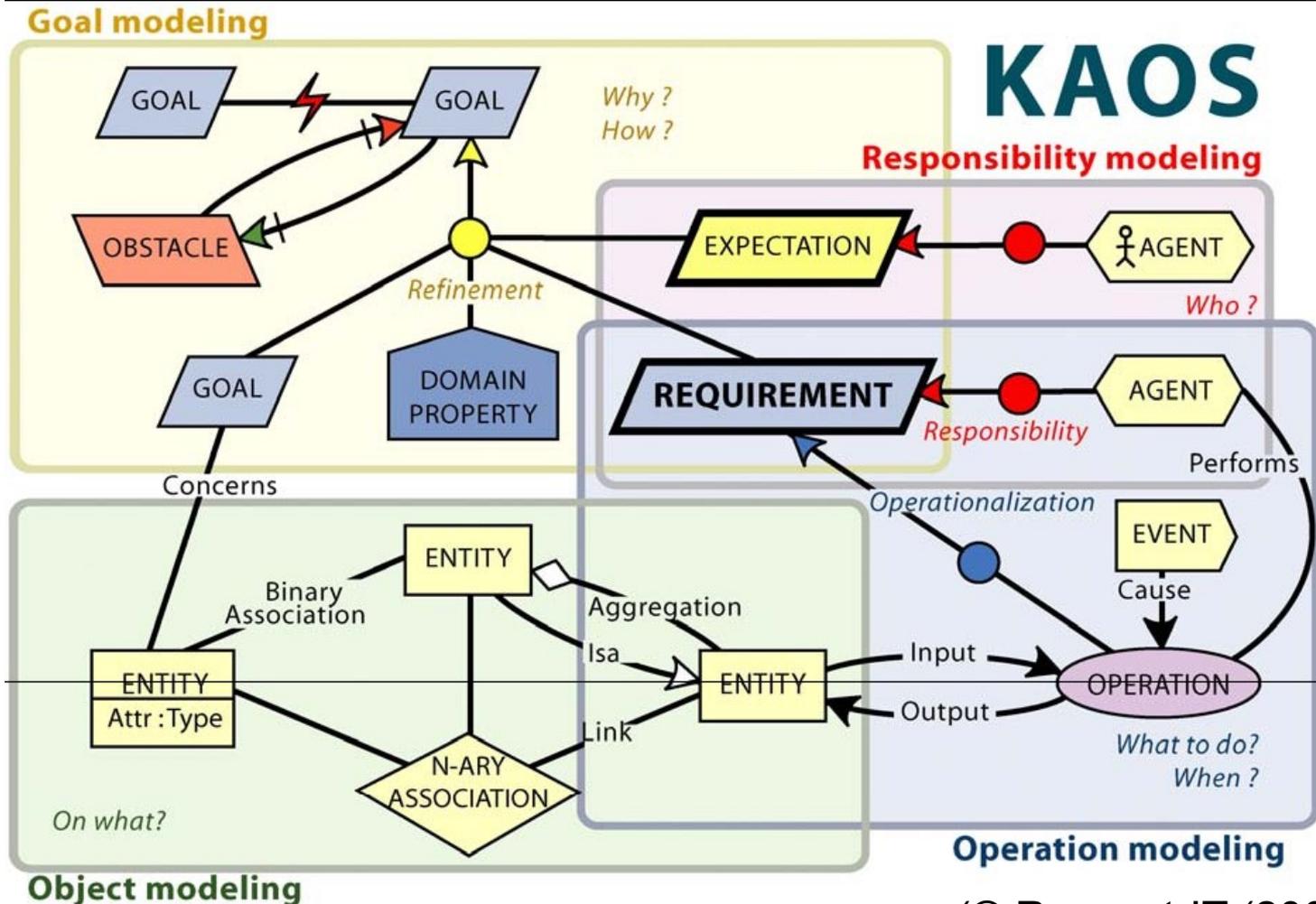


# Insider threat from sabotage

Perpetrator (who) and Motivation (why)	Stage agent (who)	Reason (why)	Method (with what)	Action (how)	Target (to what)	Immediate Effect (what)	Ultimate effect (what)
Disgruntled former employee, Ian Archer. Motivation is psychological satisfaction from revenge for perceived mistreatment	Archer. <b>1a</b>  The targets unwittingly help to give unauthorized access to Archer  <b>1b</b>	To gain system access	Social engineering  ♣ ∞ <b>1a alternative paths using email or phone</b> (arrow extends to the physical row)	Persuade or trick target to act incorrectly by giving access, setting up accounts, or giving out passwords	Security guards, system administrators, colleagues	Unauthorized physical, or logical access (via a compromised account)	Inability to produce computers → Failure to satisfy contracts → Financial losses → Reduced reputation, lost customers, lowered share price
<b>Social</b>	<b>1a</b>						
<b>Logical</b>	<b>1b</b> Archer using his account	To gain hidden logical access after termination and avoid accountability	Misuse authorized authority using own account to issue commands	Logically authorized (but prohibited by policy at the social level) commands to set up a backdoor	Network access to system	Gain unauthorized remote logical access after termination	
<b>Logical Remote</b>	<b>1c</b> Archer using a compromised account and remote access	To install malware to maintain indirect system access and avoid accountability	Use of compromised account and remote access to issue commands	Unauthorized commands to install logic bomb	Operating system of computer holding production software	Installation of logic bomb <i>Backup loss</i>	
<b>Logical Local</b>	<b>2/3</b> Logic bomb	To cause immediate damage to software → Ultimate goal to damage computer production	Privileged software misuses host system	Issue damaging commands to destroy files and software May also cover tracks by deleting log files	Software on production control system and backups → Production processes Log files	Unavailable production software → Lost computer production	
<b>Physical</b>	Archer <i>Physical attack goes from left to right cell to cell same as logical attack</i>	Render critical assets unavailable → Stop production line	Misuse allowed access to physically interfere with equipment and resources before leaving, illegitimate access after termination	Physical damage and destruction, theft, encryption	Software, backups, production computers and other essential equipment Logging devices	Damaged or unavailable systems and resources → Lost computer production	

Based on hypothetical CERT case study

# KAOS Methodology

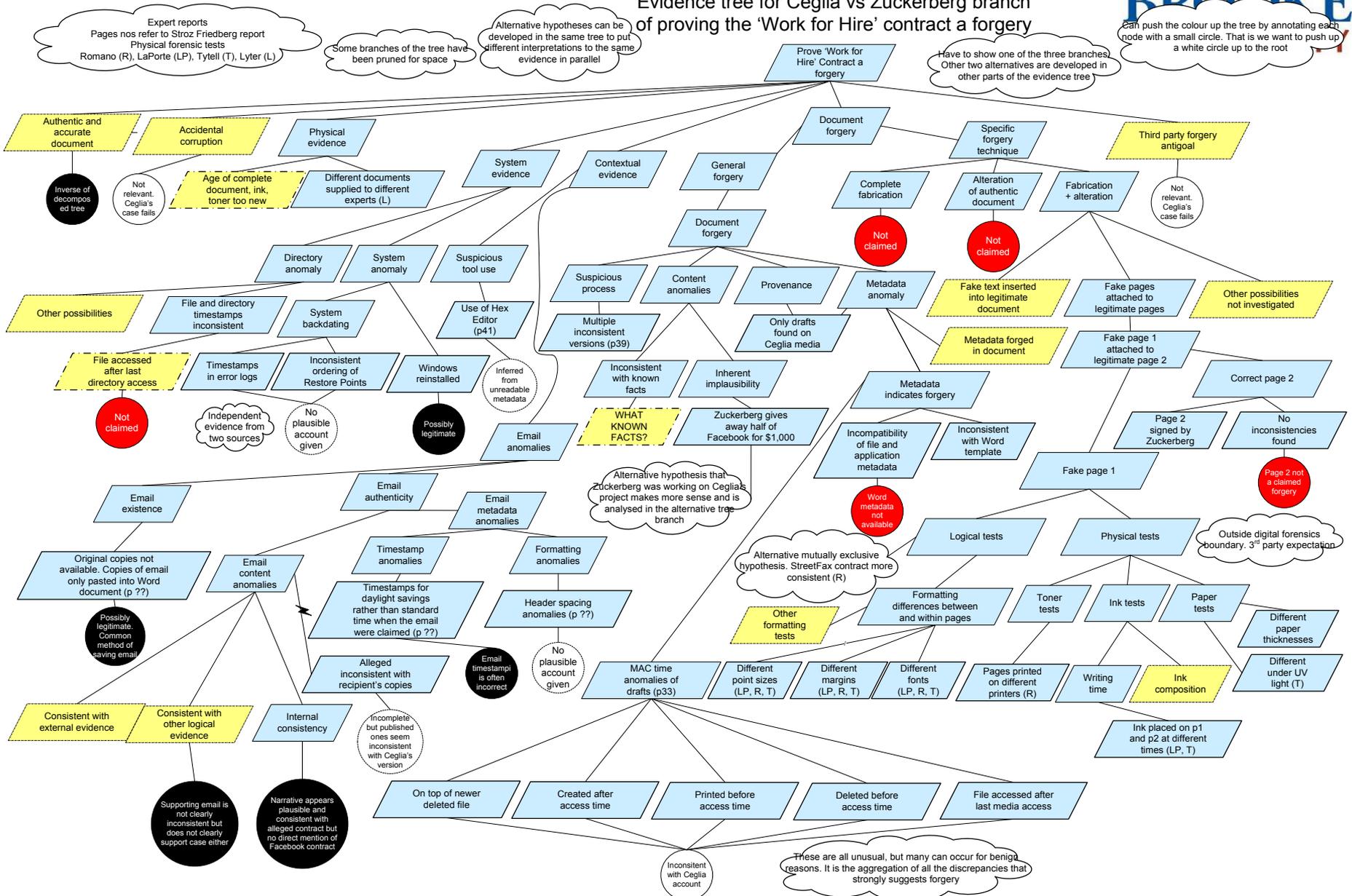


(© Respect-IT (2007))

# Ceglia v Zuckerberg and Facebook

- Paul Ceglia produced what he said was a copy of the contract he and Mark Zuckerberg had signed
- Covers two projects on which the two were working together
- Ceglia project called "StreetFax" and a Zuckerberg project called "the face book."
- Strong evidence for a contract about StreetFax, but little for a contract involving Facebook
- However, Ceglia and Zuckerberg clearly did discuss the idea in detail, and the email give some support to Ceglia's version of events
- Ceglia paid Zuckerberg \$1,000 for work on StreetFax and alleges he paid \$1,000 to fund Zuckerberg's "face book" project
- According to Ceglia, the agreement says that Ceglia will get 50% of the "face book" project in exchange for funding initial development
- Four tree branches, each of which we decompose into demonstrable statements (~ requirements) to disprove Ceglia's case
- Next slide has one branch used to prove Ceglia's contract a forgery

# Evidence tree for Ceglia vs Zuckerberg branch of proving the 'Work for Hire' contract a forgery



# Illustrative problem with signatures

- Let us say any network packet has an initial likelihood of being a particular attack with probability 1 in 1,000,000
  - And the attack always displays a certain signature
- Then, let us say any packet has this incident indicator in 1 in 1,000 cases by chance
- Now we observe a packet with the characteristic
- What is the probability it is evil?

# Illustrative problem with signatures

- Network packet has a likelihood of being bad with probability 1 in 1,000,000 and always displays a certain signature
- Any packet has this incident indicator in 1 in 1,000 cases by chance
- What is the probability that a packet with the characteristic is evil?
- Within 1,000,000 packets, 1000 would have the trait by chance + attack packet = 1001 packets
- Probability of evilness is only increased to 1 in 1001 or about 0.1%
  - 1001 (1000 + 1) packets display the signature with only one being an attack
- Conclusion: May be inundated with false positives from signatures even if bad behaviour always has this characteristic
- Multiple different indicators may be better for diagnosis obtained from the wider context provided by patterns

# Patterns in a 'science of security'

- Where are the materials, methods, experiments and conclusions characteristic of scientific endeavour?
  - 'You cannot manage what you cannot measure'
- Science based on patterns that help explain how the world operates
  - Forming and testing hypotheses for their occurrence
- Same principles apply to logical world
- Science based on descriptive patterns
  - Distinction between prescriptive and descriptive patterns (Barnum)
- Use attack patterns to explain possible incidents
  - Scientific process of hypothesising attacks and testing for evidence
  - Aids search for and recognition of incident indicators in large amounts of data and complex incidents

# Pattern matching

- Need to understand patterns to operate in world
  - Predictable and regular patterns form basis for understanding world and deciding effective actions to achieve goals
- More to recognising attacks than recognising byte strings
  - Struggling to move beyond signatures because of the problem of false positives
- Brittle failure modes if matching is too specific as adversary can adapt more quickly
  - Malware writers can test their malicious code against anti-malware until it is not detected (arms race we cannot win)
- Need to understand patterns of behaviour
  - Polymorphic and metamorphic viruses can change their form
- Hackers are much more unpredictable than malware
  - Advanced persistent threat and hacktivism increasing issues

# Logical control

- Observations taken to infer patterns from a mass of data or complex incidents like the ones shown earlier
  - Use guidance from patterns
- Amount of data is the problem
  - Incomplete, inconsistent, incorrect or excessive data
  - Look for combinations of factors that are rarely seen for benign reasons
- Bottom-up as opposed to top-down with patterns
- Using patterns directly is top-down
  - Can hypothesise and possibly test possible causes
  - Ceglia incorrect email timezone setting has benign explanation
- Two approaches complement each other
  - Identify and extract attack features to map to attack patterns

# Benefits of patterns

- Standardisation of incident expression provides solid foundation for discovery, understanding and communication of patterns
  - Simple ad hoc incident descriptions are imprecise and ambiguous
- Greater precision provides basis for reasoning and tool development
  - Theoretical model of attack and defence in OWL can reason using description logic
- Widen the scope so that defences can be deployed holistically within a wider system context
  - Wider analysis of how, when and where to respond
- May help provide more robust system with defence in depth
  - Atomic attack symptoms may change to evade detection
- Helps understand patterns of adversarial behaviour and infer unobserved characteristics such as motivation
  - Inadequate to respond tactically to intractable issues such as the advanced persistent threat (APT)
- Aid to understanding range of attack variations and new possibilities
  - Elaborate all attack possibilities proactively as aid to testing system

# Composition of patterns

- Design pattern composition has been formalised
  - Hong Zhu and Ian Bayley (PATTERNS best paper, ACM TOSE etc)
- Need new composition operators for security
- Can attack patterns be rendered harmless by composition with a defence or inverse ‘pattern’ ?
  - Pattern  $\circ$  inverse ‘pattern’  $\rightarrow$  malicious behaviour rendered harmless
  - Inverse ‘pattern’ may be a set of defensive measures
- Composition of two security patterns
  - Need to show the same security properties hold for the composite as the parts
- Decomposition of attack patterns into basic components is the reverse of composition
  - May make attacks easier to analyse and counter

# New research directions

- Demonstration of utility of patterns
  - Need good cases studies showing the benefits and potential of patterns
  - Surveys of field to understand diversity of pattern use
  - Need researchers with different skill sets here such as socio-technical people
- Relationship between theory of patterns and empirical use of patterns in industry
  - Differences in understanding of patterns between research and industry
  - Better understanding of relationship between design patterns and pattern matching in operational systems
  - Understanding benefits and application of potential novel techniques such as from AI to real problems

# Practical directions

- Where are the quick wins?
- Possible better understanding and remediation of common attacks
  - Buffer overflow and SQL injection etc
- Can write recognising automaton for common attacks
  - Determine necessary conditions for attack avoidance, detection and reaction
- Improve pattern uptake
  - Integration into tools for both detecting existing patterns and implementing new patterns
- New domains where they can usefully be applied
  - Digital forensics, cloud, critical infrastructure, cyberwarfare, IoT, socio-technical

# Future roadmap: Challenges

- Need language or ontology with stronger theoretical basis
  - Formal pattern language for attack and security patterns (in OWL)
- Need a clear story or narrative for patterns
  - Possible link to the so-called ‘science of security’ for which patterns of all kinds are fundamental
- Wider and deeper dissemination needed
  - Need for collections of case studies, surveys of field
  - Establish pattern repository, possibly a Wiki
- Need better understanding of the link between design patterns and pattern matching
  - Must move beyond signatures
  - Brittle failure if matching on specific signatures as adversary can adapt more quickly
- Tool support for discovering and implementing patterns

# The future

- More workshops planned
- Possible research council funding in UK
  - Already promised 100% funding from industry
- Establishing steering group to plan roadmap
- Establishing network of interest to which anyone can subscribe (email CBlackwell@brookes.ac.uk)
- 2<sup>nd</sup> Cyberpatterns workshop
  - 1<sup>st</sup> Cyberpatterns workshop proceedings available at <http://tech.brookes.ac.uk/CyberPatterns2012>
- Possible 1 day 'patterns in practice' workshop funded by industry
  - Industry engagement to link theory and practice

# Acknowledgements

- Oxford Brookes for funding
- Sophos for sponsorship
- Hong Zhu and Ian Bayley for initial ideas, discussion and organisation
- David Duce for writing up the proposal and workshop organisation
- Admin staff working behind the scenes to make sure the workshop ran smoothly
- Sean Barnum and Kevin Lano for inspiring keynote talks
- Programme committee for reviewing papers
- Presenters and attendees for enlivening talks and debate
- Software Assurance Forum, Dept of Homeland Security and MITRE for invitation
- Mistakes are my own responsibility

# Cyberpatterns Proceedings and Invited Contributions

# Formal Methods

# Metric-guided selection of patterns

Kevin Lano

- Semi-automate selection of patterns by defining quality measures of software design
  - Critical values of a measure indicate that one or more patterns could be applicable
- Introduction of a pattern should improve the measure, and not affect semantics (ie functionality)
- Concept has been used for model transformation patterns should be generally applicable to other pattern categories

Design Patterns for Model Transformations, Lano et al,  
ICSEA 2011

Invited contribution

# Towards A General Theory of Patterns

Hong Zhu

As knowledge of solutions to recurring design problems, a large number of software design patterns (DP) has been identified, catalogued and formalized in the past decades. Tools have been developed to support the application and recognition of patterns. However, although the notions of pattern in different subject domains carry a great deal of similarity, we are in lack of a general theory that applies to all types of design patterns. This paper is based on our previous work on formalization of OO DPs and an algebra of pattern compositions. We propose a generalization of the approach so that it can be applied to other types of DPs. In particular, a pattern is defined as a set of points in a design space that satisfy certain conditions. Each condition specifies a property of the instances of the pattern in a certain view of the design space. The patterns can then be composed and instantiated through applications of operators defined on patterns. The paper demonstrates the feasibility of the proposed approach by examples of patterns of enterprise security architecture.

# Challenges For a Formal Framework of Patterns

Ian Bayley

This article discusses the possibility of formalising patterns for cybersecurity building on previous successes in formalising design patterns, and focussing on the possibilities of applying the same techniques to attack patterns and security patterns. This would be enhanced by a common framework of patterns.

# A Strategy for Formalizing Attack Patterns

Clive Blackwell

We have created a framework for modeling security that divides computer incidents into their stages of access, use and effect. In addition, we have developed a three-layer architectural model to examine incidents with the social, logical and physical levels. Our ontology that combines the architectural and incident models provides the basis for a suitable semantics for attack patterns, where the entities and relationships between them are accurately defined and understood. The current informality of these patterns means that their utility is limited to manual use, so we plan to extend existing work on formalizing design patterns to attack patterns, to aid the automated creation of effective defensive controls. A specification in logic, which is progressively refined into code, is a common method of developing high integrity and secure software, but there are additional issues in system protection, as there are several defensive controls rather than a single program. The attack patterns form a logical specification, which is intersected with the model of the defense to determine the corresponding defensive patterns to counter the attacks. This would allow better reasoning about possible defensive response measures, and holds out the possibility of proving security against certain attacks. We outline a roadmap for formulating attack patterns in our ontology and then translating them in logic, and will go into more detail in following work.

# A pattern-based approach to formal specification construction

Xi Wang and Shaoying Lui

Difficulty in the construction of formal specifications is one of the great gulfs that separate formal methods from industry. Though more and more practitioners become interested in formal methods as a potential technique for software quality assurance, they have also found it hard to express ideas properly in formal notations. This paper proposes a pattern-based approach to tackling this problem. In the approach, a pattern system is defined in advance where each pattern describes the corresponding formal expressions for specific functions expressed in semi-formal notations, which enables the development of a supporting tool to automatically guide one to gradually formalize the specification. We take the SOFL notation as an example to discuss the underlying principle of the approach and use a specific example to illustrate how it works in practice. Our contribution in this paper is expected to set up a foundation for an automated support in the future.

Invited contribution

# Management

# Tool-supported Premortems with Attack and Security Patterns

Shamal Faily, John Lyle and Simon Parkin

Security patterns are a useful technique for packaging and applying security knowledge. However, because patterns represent partial knowledge of a problem and solution space, there is little certainty that addressing the consequences of one problem won't introduce or exacerbate another. In this abstract, we suggest that rather than using patterns exclusively to explore possible solutions to security problems, we should use them to carry out a *premortem* on why they instead cause problems. We present the approach taken to devise and tool-support such a process using data from the EU FP 7 *webinos* project.

# Management Patterns for Network Resilience: Design and Verification of Policy Configurations

Alberto Schaeffer-Filho, Andreas Mauthe, David Hutchison and Paul Smith

Computer and communication networks are becoming increasingly critical in supporting business, leisure and daily life in general. Thus, there is a compelling need for resilience to be a key property of networks. The approach we present in this paper is intended to enable the specification of management patterns that describe the dynamic intrusion tolerant behaviour of resilient networks. A management pattern describes a policy-based collaboration between a set of resilience mechanisms used to address a specific type of challenge. Much of the existing work on security patterns has focused only on the static defence aspect of a network. However, dynamic behaviour adds a great deal of complexity to network management, thus making the specification of patterns for this very desirable.

# Security Design Patterns in the MASTER Workbench

Paul J. Kearney, David A. Sinclair and Sebastian Wagner

This paper describes pattern-related aspects of the prototype Protection and Assessment Workbench developed as part of the MASTER EU 7th Framework collaborative research project. The Workbench supports a model-driven design process within the overall MASTER methodology. It includes a Protection and Regulatory Model (PRM) tool that is a step towards turning the Workbench into an ‘organisational memory’ for design practices that accumulates and improves over time. PRMs are essentially control process design patterns that incorporate proven strategies in a re-usable form, saving time and improving quality and consistency.

# Architecture

# Dynamic Monitoring of Composed Services

Muhammad Asim, Bo Zhou, David Llewellyn-Jones, Qi Shi,  
Madjid Merabti

Service-Oriented Architectures (SOAs) are becoming a dominant paradigm for the integration of heterogeneous systems. However, SOA-based applications are highly dynamic and liable to change significantly at runtime. This justifies the need for monitoring composed services throughout the lifetime of service execution. In this paper we present a novel approach for monitoring services at runtime and to ensure that services behave in compliance with a pre-defined security policy. Services are defined using BPMN (Business Process Modelling Notation) processes which can then be monitored during execution.

# What really matters in a security architecture? A new meta-model for systems and cyber security

Susan Appleby, Chris P and Colin Carter

The discipline of designing a security architecture is relatively immature and is generally lacking in foundational principles. Although there is a large and growing field of expert practitioners, they vary in approach and there is a tendency to over focus on security products at the expense of examining security aspects of the overall system. In practice there are many factors which contribute to the overall security posture of a system, but these are not generally documented or modelled explicitly.

# A Heuristic Approach for Secure Service Composition Adaptation

Bo Zhou, David Llewellyn-Jones, David Lamb, Muhammad Asim, Qi Shi, Madjid Merabti

Secure adaptation of service composition is crucial for service-oriented applications. An effective adaptation method must improve a composition's adherence to specified behaviour, performance and security guarantees at reasonable cost in terms of computing complexity and time consuming. This paper discusses current techniques that have been developed to help achieve secure service composition. Based on security verification results, which have been categorised into four patterns, a simple heuristics-based adaptation strategy is proposed. In order to make direct comparisons of different services, a novel quantification method is also introduced.

# Socio-technical Systems

# Towards a Simulation of Information Security Behaviour in Organisations

Martin Ruskov, M Angela Sasse and Paul Ekblom

In this paper we propose the fundamentals of a design of an exploratory simulation of security management in a corporate environment. The model brings together theory and research findings on causes of information security risks in order to analyse diverse roles interacting through scripts. The framework is an adaptation of theoretical and empirical research in general crime prevention for the purposes of cybercrime. Its aim is to provide insights into the prerequisites for a more functional model.

# Patterns of Information Security

## Postures for Socio-Technical Systems and Systems-of-Systems

Tim Storer, Karen Renaud and William Bradley Glisson

The paper describes a patterns approach to the challenges of engineering system-of-systems and socio-technical scale systems. The development of methods and practices which address engineering problems at the socio-technical and systems-of-systems scale remains a considerable challenge. Existing practice is largely based on craft and the instincts and experience of project engineers. Problem areas include design predictability, testing and (critically) maintenance and evolution of systems in the presence of rapid and concurrent environmental and system change. A patterns approach has been successfully applied in other areas of software engineering in order to rationalise and codify engineering knowledge and best practice. We believe that the use of patterns formulated around the responsibilities held by actors within organisations can be successfully used to address the challenges of engineering secure large scale systems of the future.

# Pattern matching

# An Overview of Artificial Intelligence based Pattern Matching in a Security and Digital Forensic context

Faye Mitchell

Many real world security and digital forensics tasks involve the analysis of large amounts of data and the need to be able to classify parts of that data into sets which are not well or even easily defined. Rule based systems can work well and efficiently for simple scenarios where the security or forensics incident can be well specified. However such systems do not cope as well where there is uncertainty, where the IT system under consideration is complex or where there is significant and rapid change in the methods of attack or compromise.

Artificial Intelligence (AI) is an area of computer science that has concentrated on pattern recognition and in this extended abstract we highlight some of the main themes in AI and their appropriateness for use in a security and digital forensics context.

# Malware Capture & Analysis

Renato Cordeiro de Amorim and Peter Komisarczuk

- Capture HPC high interaction honeyclient
  - Drive-by-download and client side data gathering
  - Detailed logs of activity (network, process, file, etc.)
  - Now emulates all ActiveX Content
  - HoneyNet Project and Victoria University of Wellington
- Use Caputre' s log files to create malware signatures/patterns based on their behaviour
  - Recent work is based on a partitional clustering method
  - Clustered around 8,500 malware per minute (20 x faster than other techniques published)

# Attack Pattern Recognition Framework

Noor-Ul-Hassan Shirazi, Alberto E. Schaeffer-Filho and David Hutchison

- We intend to show:
  - How to analyse multi-stage attacks from different viewpoints revealing meaningful patterns with respect to various attack features.
  - How to combine systematically all viewpoints such that behaviour properties of attacks are modelled.
- We assume that:
  - Coherent data sets are available for analysis under assumption that high degree of coordinated multi-stage attacks should be reflected by various correlation patterns between attack events
- Research Questions:
  - How can we identify a correlation of data that corresponds to an attack, based on the combination of all available evidence?
  - How can we analyse the cyber-attack phenomenon from separate viewpoints, revealing different correlation patterns?
  - Can we develop a model that fits multiple datasets and not just one specific dataset?
  - How to device a meta-model that is suitable for all /some of the attack patterns?

# Experimental Techniques and Measurement

Les Hatton

- Software engineering has not been a critical discipline and is generally characterised by a lack of measurement. However patterns are the essence of measurement and analysis whether Cyber derived or otherwise. It will be very important for the development of this nascent area that good empirical principles are founded and patterns defined by solid statistical rigour. Only then will we be able to exploit those patterns reliably.  
In some areas, data analysis is relatively unambiguous and noise-free posing a relatively simple problem to handle. Such is generally the case in the identification of e-mail born phishing attacks for example, which can be identified with six sigma accuracy. In other areas such as some kinds of intrusion attack, or identifying viral strains, can present a considerable challenge. However, provided we face this with the formidable armoury of statistical analysis techniques and the very considerable experience in the design of suitable experiments which has been accumulated over the years, there seems every hope that the study of Cyberpatterns, if not putting us ahead of the game, should make us a difficult proposition for any adversary.

Invited contribution

# Applications

# Where has this hard disk been?

## Extracting geospatial intelligence from digital storage systems

Harjinder Singh Lallie and Nathan Griffiths

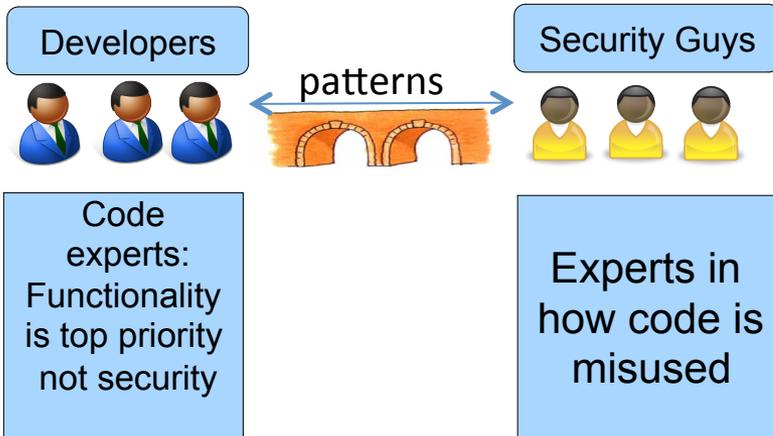
Digital storage systems (DSS) contain an abundance of geospatial data which can be extracted and analysed to provide useful and complex intelligence insights. This data takes a number of forms such as data within text files, configuration databases and in operating system generated files - each of which require particular forms of processing. This paper investigates the breadth of geospatial data available on digital storage systems, the issues and problems involved in extracting and analysing them and the intelligence insights that the visualisation of the data can provide. We describe a framework to extract a wide range of geospatial data from a DSS and resolve this data into geographic coordinates.

# Extending AOP Principles for the Description of Network Security Patterns

David Llewellyn-Jones, Qi Shi, Madjid Merabti

Aspect Oriented Programming is increasingly being used for the practical coding of cross-cutting concerns woven throughout an application. However, most existing AOP point-cut definition languages don't distinguish in their application between different systems across a network. For network security there is a need to apply different aspects depending on the role a piece of code has within the larger networked system, and a new approach for this is therefore required. In this paper we present a formalism for how this might be approached, proposing a way to capture distributed point-cuts for applying different aspects in different parts of the network. The method is based on templates that match properties within the code, and a set of flexible relationships that can be defined between them.

## Knowledge Gap



## Consider this code:

```
int x = atoi(argv[1]);
char src[64];
char dest[64];
if (x < sizeof(dest)) {
memcpy(dest, src, x);
} else error();
```

Looks secure?

What if x is negative? Not secure!

- Our patterns bridge the gap between developers and security experts
- Patterns make detailed knowledge on code misuse available to the developers

## An example pattern:

Name	1. Origin	2. Behaviour	3. Criteria Required	4. Impact	5. Countermeasures
Memcpy() Overflow	Originates where the number of bytes to be copied into *dest is greater than the sizeof(*dest).	A write operation exceeds the target memory allocated for it, and overwrites core variables on the stack such as ret or the base pointer.	Requires that either: - Attacker controls size of *dest, and contents of *src - Attacker controls content of src and an integer > sizeof(*dest) is used as an argument to memcpy()	Arbitrary code execution and corruption of stack variables.	Ensure that the number of bytes to be copied is not greater than sizeof *dest prior to making the function call.

# Composition Patterns of Hacking

Sergey Bratus, Julian Bangert, Alexandar Gabrovsky, Anna Shubina, Daniel Bilar and Michael E. Locasto

You do not understand how your program *really* works until it has been exploited. We believe that computer scientists and software engineers should regard the activity of modern exploitation as an applied discipline that studies both the actual computational properties and the practical computational limits of a target platform or system. Exploit developers study the computational properties of software that are not studied elsewhere, and they apply unique engineering techniques to the challenging engineering problem of dynamically patching and controlling a running system. These techniques leverage software and hardware composition mechanisms in unexpected ways to achieve such control. Although unexpected, such composition is not arbitrary, and it forms the basis of a coherent engineering workflow. This paper contains a top-level overview of these approaches and their historical development.

# Testing

## Testing

Dianxiang Xu

- Security Testing Using Attack Patterns
  - STRIDE: Spoofing, Tampering, Repudiation, Info Disclosure, DOS, Elevation of Privilege
- Testing of Security Policies and Mechanisms
  - Role-based access control
  - Test model = functional model + access control rules
- Tool Support: MISTA (Model-based Integration and System Test Automation)
  - <http://www.homepages.dsu.edu/dxu/research/MISTA1.0.zip>

# Using Attack Patterns to Develop a Penetration Testing Framework

Clive Blackwell

We show that both the automated tool and skill-based methods of penetration testing are unsatisfactory, because we need to provide comprehensible and complete evidence to clients about their weaknesses, and offer an adequate remediation plan to fix the critical ones. We use attack patterns to suggest a penetration-testing framework to help avoid the limitations of current approaches. There are further benefits in formalising attack patterns, which we propose to develop in future work.