Cyber Security Control Effectiveness

Automated Analysis of Threats and Countermeasures for SMEs

Author Information Redacted

***Abstract*—This paper describes the use of machine understandable modelling techniques to characterise threats based on local or remote exploitation of software vulnerabilities, and subsequent machine reasoning to evaluate the effectiveness of a well-known control strategy (the UK Cyber Essentials scheme) in counteracting those threats for typical SMEs. The machine understandable modelling techniques were taken from the FP7 OPTET project, and were used to classify 'well-known' threats documented in the Mitre CVE database. This led to new insights into the limitations of such vulnerability oriented databases, and a proposal for a new classification approach. SME network models were then created based on interviews with local SMEs, and machine reasoning used to identify which threats could affect those networks, and which could be counteracted by using Cyber Essentials. The conclusion was that Cyber Essentials is quite effective for traditional SMEs with well-defined perimeters, but less so for SMEs that make significant use of cloud services and/or mobile connectivity for remote workers.**

Keywords—cybersecurity, threat analysis, machine reasoning

I. Introduction

It is widely accepted that SMEs find it difficult to respond to cyber security threats. A UK Government survey of 4300 small businesses conducted in 2014 [1] examined all aspects of business operations. It established that:

- 82% of small businesses have fewer than 10 employees, and only 3% have 50 or more employees;
- 88% operate from a single site, often the owner's home;
- 98% of small businesses make use of the Internet as a key part of their operation, and 77% operate their own website;
- 89% of small business employees were able to access their employer's IT systems from home, and 74% had access via smartphone;
- 28% use contract staff and hence have a relatively high turnover of users in their IT networks.

These facts show that SMEs are as dependent as any other business on IT systems and the Internet. They should therefore be as concerned about potential disruption of their business from cyber threats. However, only 61% of SMEs surveyed claimed they are good at adopting industry best practice, and only 13% have any provision for IT training at managerial level. Interestingly, the survey did not mention cyber security issues in the questions about potential barriers to business success, suggesting that SMEs were not expected to be very aware of the impact of cyber security breaches. This was probably justified as when seeking business advice from external sources, only 2-3% of requests from SMEs are IT or technology related.

In June 2014, the UK launched the Cyber Essentials scheme [2] to provide guidance and standards aimed at SMEs to defend themselves from cyber attacks. The main provisions of the Cyber Essentials cover the use of boundary firewalls and internet gateways, secure configuration of IT systems, user access control, malware protection and software patch management. An accompanying specification defines an assurance framework for Cyber Essentials implementation, and several accreditation providers in the UK now support certification. Awareness of Cyber Essentials among SMEs remains quite low, even though Cyber Essentials certification has been mandatory for suppliers to the UK Government for contracts that involve handling sensitive or personal data.

Cyber Essentials is deliberately formulated as a lightweight approach to cyber security, in recognition that SMEs will be unable or unwilling to implement anything else. It aims to deal with around 80% of common cyber threats [3]. Of course, SMEs who may be ignorant or cynical about cyber threats are unlikely to consider implementing measures unless they do provide significant protection against business risks. The authors therefore set out to test whether Cyber Essentials lives up to its own billing, using a novel analytical approach for identifying threats and countermeasures using semantic modelling and machine reasoning methods.

II. Related Work

The most widely used standard for security verification is *ISO 15408*, also known as *Common Criteria* [4]. This defines procedures for assessing security properties (claims) to different evaluation assurance levels (EALs) from EAL1, which involves simple testing, up to EAL7, which involves the use of formal verification. Common Criteria only verifies whether security measures have been correctly implemented. It does not provide any assurance that those measures appropriately manage the risks to which the system is exposed. Also, users must identify and analyse risks to determine what security measures are actually needed. The most widely used standards for this are *ISO 27001* [5], which defines information system risk management requirements, and *ISO 27005* [6], which specifies how to identify threats, i.e., the sources of risk. *Cyber Essentials* itself does not involve any analysis of threats, as it simply specifies a set of security measures that should be used. However, to determine how effective these measures will be for a given SME, one must enumerate threats involving the SME's IT network, and then find out how many of these threats are

Acknowledgement of support – to be added in non-anonymous version.

addressed by the security measures mandated by Cyber Essentials.

Cyber security threat identification and analysis remains a notoriously difficult process. Three approaches are commonly used:

- *attacker-centric* methods start by considering who might want to attack a system, and how they might do so;
- *software-centric* methods look for weaknesses in the system software, through which the system could be attacked;
- *asset-centric* methods start by considering how the assets in the system may be compromised and with what effects.

Each approach has its own strengths and weaknesses, and its own supporting tools and methodologies. Software centric methods are actually best supported in terms of tools, because they focus on the analysis of software code which can be conducted using automated methods to detect common flaws. There are several levels at which risks can be classified. One can identify threats based on exploits, as in Microsoft's *STRIDE* [7], or by rating the threats as in *DREAD* model [8]. One can also assessed threats through development phases, starting from the requirements, using methods such as *SQUARE* [9], or during the design phase using e.g. *UMLSec* [10]. Threats can also be assessed on the finished component using tools like *Commix* [11]. The problem with software centric methods is that they focus on one type of weakness, and do not consider other factors, such as human ones, or systemic weaknesses that may allow the correct functionality to be misdirected or otherwise abused.

Attacker centric methods do better in these respects. Attack trees [12] can be created by identifying attack goals, to analyse the security of systems and subsystems, supported by tools such as *SecurTree* [13], where attacks are modelled as a tree of actions or events leading to compromise. Such tools help analysts keep track of possible sequences of actions, and in some cases quantify the effect of security measures in reducing the ability of attackers to carry out those actions. Another approach is to profile attackers' behaviour and analyse it to determine appropriate responses. One example is profiling attackers' attempts to log into systems in order to compromise their security policies [14]. One can also analyse attack traces to characterise attacker behaviour [15]. The attacker centric approaches can also benefit from resources such as attack patterns published in databases like *CAPEC* [16]. However, attacker centric methods still depend on the analyst's ability to identify attackers and potentially damaging actions or events. Any errors may lead to a significant number of possible threats being overlooked.

It is for these reasons that ISO 27005 mandates the use of an asset centric risk analysis approach. However, this too depends critically on the expertise of the analyst. Tools such as *Trike* [17] and *OCTAVE Allegro* [18] provide a framework for the analysis, but the only support for comprehensive coverage is provided by checklists describing potential concerns. One major issue for asset centric methods is the need for expert knowledge of both the system being analysed and IT security state of the art: only then is it feasible to provide a complete analysis of system assets and the effect of compromise on the system, and ways in which assets might be compromised or protected. OCTAVE Allegro provides checklists to support its use by system analysts who may lack IT security expertise, but in many cases one must use IT security consultants to carry out a credible analysis of threats. The main concern then is to ensure that system and security experts share their respective knowledge effectively so nothing is overlooked through miscommunication. For SMEs, another concern is the cost of IT security consultants, which may prevent the use of asset centric risk analysis altogether.

Semantic modelling of security issues was first proposed as a way to communicate more precisely about IT security, thus reducing the risk of miscommunication between experts. A useful review of early efforts in this direction is provided by Blanco, et al [19]. One of the most successful of such efforts was demonstrated by Secure Business Austria, which created a semantic encoding [20] of the German IT Grundschutz Manual [21], a system-level checklist of potential IT security threats and vulnerabilities. More recently, this approach has been extended, creating a generic threat classification rather than a descriptive schema. The ontology is based on the SBA core structure, but this is simplified to reduce the number of facts that must be asserted before new insights can be found by machine reasoning. Classes of threats are defined in terms of their relationships to interacting asset types, making it possible to automatically map these threats onto a model of a system composed of those asset types. The approach was developed in the FP7 SERSCIS project where it was used for run-time classification of disruptions to airport information exchanges [22], and refined in the FP7 OPTET project where tools were developed to perform automatic threat identification and analysis at design time [23].

The semantic modelling approach has several advantages for the analysis of Cyber Essentials. Firstly, a single '*generic*' model can be created describing potential threats and control strategies, which can then be applied easily and automatically to several different system models. This is important because SMEs use a range of different approaches when provisioning their IT infrastructure, so it was necessary to analyse several distinct network architectures with respect to the same set of threats. Also, provided that the knowledge base has sufficient coverage, the threat catalogues resulting from the analysis will be comprehensive, since the procedure is fully automated and therefore not subject to occasional human error that may lead to some threats being overlooked. Finally, because IT security knowledge is encoded in the generic model and kept separate from system modelling, the approach can easily combines expertise in both areas in successive phases with no risk of oversight caused by miscommunication.

Of course, the success of this approach does depend on a security knowledge base that provides a comprehensive threat catalogue. In general, threats are categorised into three main types; *accidental*, *deliberate* and *environmental* [24]. For the purpose of this study the most important category are potential deliberate threats, including those involving exploitation of vulnerabilities that are being discovered all the time [25]. There are two main sources of data for these threats: databases such as the '*National Vulnerability Database*' (NVD) and the '*Common Vulnerabilties and Exposures*' catalogue (CVE), and reports generated by analysts such as national *CERT* teams and the

'*Open Web Application Security Project*' (OWASP). The CVE database [26] provides the most comprehensive open access data about each vulnerability, and it was taken as a reference point and used to expand an semantic knowledge base published by the FP7 SERSCIS project [27].

III. Common Threats Identification

The CVE database has a significant number of vulnerabilities reported each year. The CVE database statistics list 69402 vulnerability entries on the 30th March 2015. Each entry has a CVE entry number, description of the vulnerability, publishing and modification date and references to other information sources. Manual mapping from CVE to semantic models is challenging because of the vast number of CVE entries to analyse. Moreover, the description for each entry is unstructured text which makes automated analysis difficult. In practice, it turned out that the schema used in the CVE database is also incompatible with an asset centric approach to modelling threats. Nevertheless, by using CVE data as a starting point, it was possible to ensure comprehensive coverage of the vast majority of threats.

To do this, a three step procedure was adopted, as shown in Fig. 1.

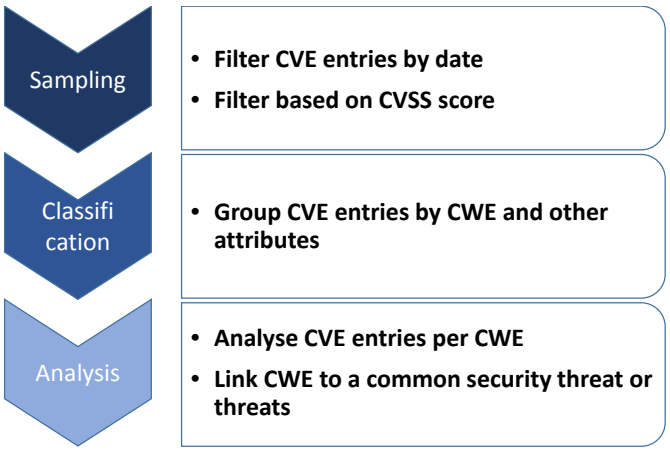

Fig. 1. Identifying common threats from CVE data

The first step was designed to reduce the number of CVE entries for analysis by sampling the data. By considering only entries added in a two year period starting 01 March 2013, the number was reduced to from nearly 70,000 to 1,016. Then since our interest was common threats that could significantly disrupt an SME business, we filtered again based on the Common Vulnerability Scoring System (CVSS) score. This is way to quantify the severity of vulnerabilities based on what access is needed, the difficulty of the attack, and the impact. We focused on threats with a CVSS score of at least 6.8, which includes local attacks that are low complexity (i.e. easy to carry out), and lead to a complete compromise of confidentiality, integrity or availability. More complex or less compromising attacks score above this threshold if and only if they can be executed remotely. By filtering on CVSS score, the number of CVE entries retained for further analysis was reduced to 494.

The next step was to organise the CVE entries into groups that could potentially be mapped to specific semantic threat classes. This was done by using the Common Weakness Enumeration (CWE) scheme, a schema for vulnerability types. The National Vulnerability Database (NVD) provided by NIST provides a link between CVE references and CWE classes, making it easy to perform this grouping. The CWE entries considered most relevant were as shown in TABLE I.

TABLE I. Common Weakness Enumeration Entries

CWE	Description
79	Failure to Preserve Web Page Structure ('Cross-site Scripting')
119	Failure to Constrain Operations within the Bounds of a Memory Buffer
89	Improper Sanitization of Special Elements used in an SQL Command ('SQL Injection')
20	Improper Input Validation
94	Failure to Control Generation of Code ('Code Injection')
200	Information Exposure
22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
287	Improper Authentication
59	Improper Link Resolution Before File Access ('Link Following')
362	Race Condition
134	Uncontrolled Format String
78	Improper Sanitization of Special Elements used in an OS Command ('OS Command Injection')
284	Access Control (Authorization) Issues
77	Improper Sanitization of Special Elements used in a Command ('Command Injection')
74	Failure to Sanitize Data into a Different Plane ('Injection')
345	Insufficient Verification of Data Authenticity

Investigating the CVE entries corresponding to the above CWE classes revealed that each entry actually corresponded to more than one type of exploit, as shown in TABLE II.

TABLE II. CWE Entries Linked to Common Security Exploits

Exploit	CWE Class															
	79	*119*	*89*	*20*	*94*	*200*	*22*	*287*	*59*	*362*	*134*	*78*	*284*	*77*	*74*	*345*
Code execution	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Cross site scripting	X		X													
Denial of service		X		X	X	X	X	X	X	X	X	X				
Bypass restriction	X	X	X	X		X	X	X	X	X			X			
Directory traversal	X	X				X	X	X	X							

Exploit	CWE Class															
	79	*119*	*89*	*20*	*94*	*200*	*22*	*287*	*59*	*362*	*134*	*78*	*284*	*77*	*74*	*345*
Gain privileges		X		X					X	X	X	X	X			
HTTP response splitting				X	X											
Memory corruption		X								X						
Information leakage		X		X		X	X	X	X	X			X			
Memory overflow																
Cross site request forgery	X		X				X	X								
File inclusion			X	X	X									X	X	X
SQL injection			X			X										

This table reveals once source of difficulty in using CVE data and CWE classes for design-time security analysis. There is a considerable overlap between CWE classes, e.g. many of the classes listed in TABLE I. could be seen as ultimately due to a failure to check input data. Probably as a consequence of this, there is considerable variation in the way contributors to the CVE vulnerability database choose to classify vulnerabilities. As a result, the same basic scenario appears in multiple categories. The distinction between CWE classes is sometimes based on the type of software bug leading to a vulnerability, and sometimes on the type of compromise produced by an exploit of that vulnerability. This is useful information for either software developers or forensic analysts, but it does not offer much help to a system designer or risk analyst.

If one inspects CVE entries in detail, it is also apparent that the same CWE category may be used for local or remote attacks. For example, consider two SQL injection attacks:

TABLE III. EXAMPLE SQL INJECTION CVE ENTRIES

CVE	Description	Access	CWE
CVE-2013-0140	SQL injection vulnerability in the Agent-Handler component in McAfee ePolicy Orchestrator (ePO) before 4.5.7 and 4.6.x before 4.6.6 allows remote attackers to execute arbitrary SQL commands via a crafted request over the Agent-Server communication channel.	Local Network	89
CVE-2013-3578	SQL injection vulnerability in the Help Desk application in Wave EMBASSY Remote Administration Server (ERAS) allows remote authenticated users to execute arbitrary SQL commands via the ct100$4MainController$TextBoxSearchValue parameter (aka the search field), leading to execution of operating-system commands.	Remote	89

Here we have two different access routes (using local or remote networks), and in one case access must be authenticated for the attack to succeed. In practice, when classifying these threats semantically, it does not matter whether the network is local or remote – only that access is via a network. Whether the network is indeed local is a function of the relationship between the network and the vulnerable component, and the relationship here is clearly only that the component can be accessed from the network. In fact, all that is needed is a diagram showing the components involved in the attack. The distinction between authenticated and anonymous access is however semantically significant. This is because an attack requiring authenticated access can be blocked using some additional control strategies.

To validate the semantic threat model, therefore, it was only necessary to check that threats exist that reproduce the patterns of involved components and the access levels specified in the CVE data.

IV. SEMANTIC MODELLING

A. Overview

The semantic modelling approach was developed in the FP7 SERSCIS project [28], and refined in the FP7 OPTET project [29]. Entities are described (actually classified) in terms of their relationships to each other, using a semantic encoding (RDF, OWL and SPARQL) supporting automated analysis using machine reasoning tools.

The ‘core’ model is shown in Fig. 2. The core concepts are asset, threat, misbehaviour and control. A threat is considered to be a combination of two elements: a threat action (a threatening situation, process or event through which the threat could damage a system), and the threat consequences (the damage caused by the threat). A key aspect of the approach is that the threat action is modelled in terms of a pattern of interacting assets that must be present for the threatening situation, process or event to arise. As far as we are aware, this is unique in allowing one to automatically reason about the relationship between threats and the architecture of an IT system to which they may apply. The threat consequences are modelled in terms of the misbehaviour induced in these involved assets if the threat becomes active. Finally, a means to control a threat (if any exist) is captured in terms of a control strategy. This consists of a set of security controls protecting the involved assets which may block the threat (i.e. prevent the threatening action arising among the involved assets) or mitigate its consequences (prevent the assets from misbehaving as a result of the threat).

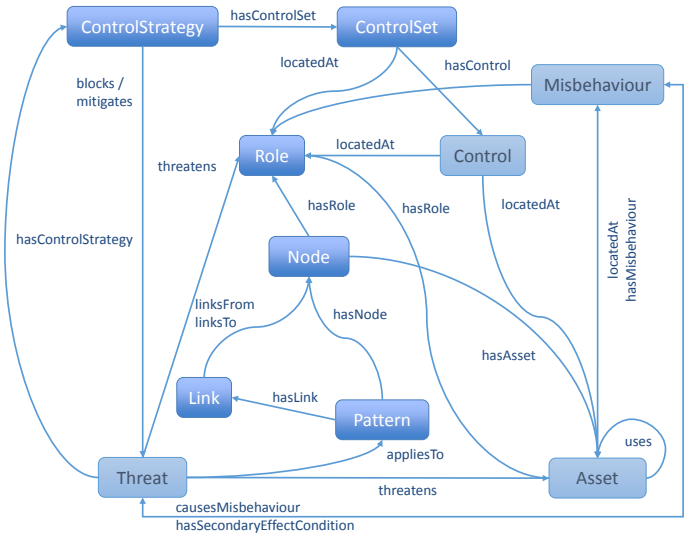

Fig. 2. Core semantic model

The core model provides the base classes for two further models:

- a '*generic*' model describing the types of assets, threats, misbehaviours and controls that may be found in systems in a given domain; and
- a '*system*' model describing an IT system in terms of more specialised asset sub-classes with specific relationships reflecting their roles in the modelled system.

The idea is that the generic model is created by a group of security experts who understand the domain of interest. In this context, a 'domain' is actually a type of IT system, e.g. a classical client-server network, a cyber-physical system, an Internet-of-Things assembly or network, etc. For the purpose of this study, we used a generic model for networks of commodity components typically used by SMEs.

The system model is then created by the system designer or security analyst, using their unique understanding of the system under consideration. Once the system model has been created, machine reasoning is used to 'compile' the system model by applying the knowledge encoded in the generic model. This produces a list of potential threats, with associated potential consequences and control strategies.

B. *Asset model*

The asset model used for this study reflects the composition of typical SME networks. The most basic classes are shown in Fig. 3:

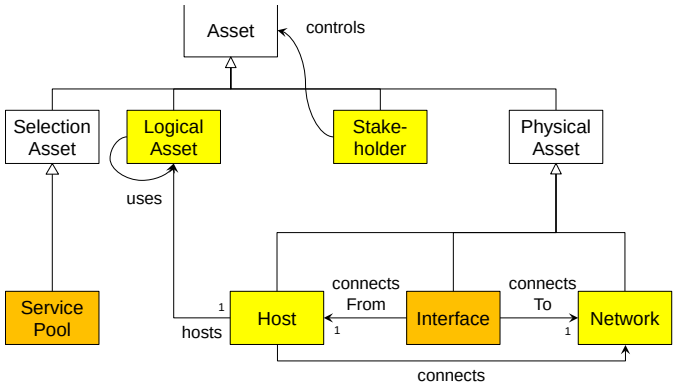

Fig. 3. Basic asset classes

The way to interpret these asset types is as follows:

- Stakeholder: a person or organisation, i.e. as asset having the will to act;
- LOGICALASSET: a process, usually implemented via software;
- PHYSICALASSET: the infrastructure or environment supporting processes, divided into:
  - HOST: an asset that provides the execution environment for a process;
  - NETWORK: an asset representing a logical communication facility allowing processes executed by different hosts to communicate with each other.
  - INTERFACE: represents the link between a Host and a Network, a possible point of control
- SELECTIONASSET: a pool of assets from which other assets can choose which one they will use.

For our purposes, a HOST is actually one or more computer systems used by SMEs to support their LOGICALASSETS. We defined subclasses representing servers, workstations, smart phones and routers (a router runs a process for managing communications, thereby defining the connectivity provided by Network assets).

The SERVICEPOOL is a SELECTIONASSET whose relationships to LOGICALASSETS capture the ways in which LOGICALASSETS can interact. There is one SERVICEPOOL for each potential interaction. The CLIENT is the LOGICALASSET that initiates such an interaction, and the SERVICE is the LOGICALASSET it interacts with. In practice there is often a pool of candidate Services, as indicated by the name SERVICEPOOL. There are two other related assets: the asset that decides which LOGICALASSETS are in the SERVICEPOOL, and the asset that decides (at runtime) which will be used. In the simplest case these last two are also the CLIENT (meaning it is in control of everything), but by varying the relationships one can represent scenarios in which the CLIENT is really a DELEGATE or AGENT.

For our purposes, it was convenient to define LOGICALASSET subclasses for specific types of processes: WEBSERVER and WEBBROWSER representing HTTP/HTML servers and clients, MAILAGENT, MAILSTORE and MAILCLIENT

representing SMTP and IMAP/POP3 servers and clients, FILESERVER which supports file sharing protocols such as SMB, a DBSERVER class representing a data storage service accessed via the SQL query language, and a REMOTELOGIN class representing a service enabling remote access to the underlying HOST.

C. Security controls

The UK Cyber Essentials scheme specifies five key control strategies:

- *Boundary firewalls* and Internet gateways to prevent unauthorised access to or from private networks;
- *Secure configuration* ensuring that systems are configured in the most secure way for the needs of the organisation;
- *Access control* ensuring that only those who should have access to systems do have access at the appropriate level;
- *Malware protection* ensuring that virus and malware protection is installed and up to date;
- *Patch management* ensuring that the latest supported version of applications are used, and that all the necessary patches supplied by the vendor are applied.

In the semantic modelling approach, a CONTROL subclass represents a requirement to implement a specific type of security measure. Each CONTROL requirement is associated with one or more ASSET types, thus modelling which security measures should be applied to which elements of the system. Each threat has zero or more control strategies describing combinations of controls that can counteract the threat. These can be used to determine which threats are or are not addressed by a given set of controls.

There is no direct mapping between the concept of CONTROL (which are related to specific system assets) and Cyber Essential control strategies (which apply to the system as a whole). For example, the OPTET ontology included a CONTROL subclass called FIREWALLRULE to represent the blocking of messages directed to or from a given Interface. However, Cyber Essentials boundary firewalls cannot be represented by adding this to every Interface. This represents a very secure but unusable system in which legitimate traffic would also be blocked. The semantic modelling approach can handle these practical aspects such as where in the network each type of control should be used. Moreover, the Cyber Essential chapter on boundary firewalls also specifies that the networking devices and control interfaces should be securely configured (i.e. default passwords changed and guest accounts deactivated) and access controlled. These are separate control concepts in the semantic model, which are also relevant to the Cyber Essentials chapters on secure configuration and access control for other devices.

For our purposes, therefore, it was sufficient to ensure that the generic model included control subclasses representing the types of asset-related security mechanisms that would be used to implement Cyber Essentials. The classes used to do this were:

- ACCESSCONTROL: represents a means to prevent an asset's functionality being used by an unauthorized user;
- ANTIMALWARE: a mechanism for detecting and disabling malicious code prior to execution by a host;
- CLIENTAUTHENTICATION: a means by which an asset can check the identity of a user attempting to initiate an interaction with it;
- IDENTIFICATION: a way for a LOGICALASSET to express its identity (or other attributes), typically via knowledge of a shared secret, e.g. username/password;
- FIREWALLRULE: represents a filtering of rules at an Interface between a HOST and a logical NETWORK, subclasses can be used for specific protocols and for inbound or outbound traffic, thus INFIREWALLALL would block attempts to create a connection from elsewhere on the NETWORK to the HOST via the INTERFACE;
- MAILSCANNING: a mechanism for identifying and warning users if emails appear to contain malicious content, e.g. obscured web addresses;
- SANDBOXING: restricting access from a LOGICALASSET to its host, e.g. limiting access to a specific area of the host's filing system, or a specific user's privileges;
- SECURECONFIGURATION: changing default administrator passwords, disabling or removing unnecessary accounts including guest accounts, and disabling auto-run;
- SECURETRANSPORT: a mechanism to prevent tampering or reading of a message in transit between two logical assets;
- SERVICEAUTHENTICATION: a means by which an asset can check the identity of a user controlling another asset with which it wishes to interact;
- SOFTWAREPATCHING: represents a systematic application of software updates to eliminate vulnerabilities;
- STRONGIDENTIFICATION: a means of identification that does not depend on a shared secret, e.g. a PKI identity certificate;
- TRUSTMANAGEMENT: means ACCESSCONTROL policies limit access to highly trusted individuals.

The last of these was added specifically to support analysis of SME networks. This is because small SMEs often use this approach, e.g. restricting access to critical systems or data to the founders of the company, or a few long-serving employees. Additional control classes were available in the original semantic control model but were not specifically relevant to Cyber Essentials strategies. They were retained in the ontology but were used only to explore some possible extensions of the Cyber Essentials scheme.

D. Threat models

The final step to complete the generic model is to define threats in terms of their relationships to assets, misbehaviours and controls. A threat is considered to be made up of two elements: a threat action representing the circumstance or event whereby the threat can arise, and a threat consequence which represents the effect of the threat on a target asset if it does arise. For our purposes (viz., evaluating the effectiveness of cyber

controls) we are not greatly concerned to distinguish different threat consequences, since any significant damage is to be avoided.

Threats were therefore modelled by considering patterns of vulnerabilities analysed in Section III, and identifying which types of assets could be involved. This was done by taking a few high-impact examples in each of the leading CWE categories, and mapping those examples onto existing or (where necessary) new semantic threat models.

For example, consider the case of SQL Injection attacks, which exploit the vulnerability of improper neutralisation of special elements used in an SQL command, designated CWE-89 in the Common Weakness Enumeration dictionary. Careful inspection of CWE-89 shows that the weakness arises due to a vulnerability in a trusted application that uses a SQL database. The vulnerability enables an attacker to pass a destructive SQL query to the database. For this to be the case, the following assets must be present:

- the vulnerable application (i.e. a LOGICALASSET);
- a back-end database server, which is used by the vulnerable application;
- a HOST, which should accept connections from a NETWORK, and which hosts the vulnerable application;
- the Interface between the HOST and NETWORK;
- a SERVICEPOOL containing the vulnerable application.

The SERVICEPOOL is included to disambiguate this threat from other threats, by ensuring that the vulnerable application is a SERVICE (i.e. a LOGICALASSET that can accept interactions initiated by others). The Interface is included because it represents a possible point of control (using a FIREWALLRULE) even though its role in the threat action is passive.

In the semantic threat ontology, this threat is known as DB.X.SQLIS-S-RA. This nomenclature captures the key elements of the threat as follows:

- which asset is threatened, i.e. the target of the attack, in this case the DBSERVER, abbreviated to a one or two letter prefix, hence 'DB';
- the type of consequences, ranging from '0' (meaning the attacker gains control of the target asset) to '12' (meaning the asset is rendered unauditable), or 'x' to represent any potentially harmful consequence;
- the name of the asset pattern depicted, in this case SQLIS, meaning the assets involved in an SQL injection attack via a Service;
- the vulnerable asset, in this case the LOGICALASSET acting in the role of Service, hence 'S';
- the type of attack, represented by R or L indicating whether the attack is remote or local;
- the type of attacker, represented by A or U indicating if they are anonymous or an authenticated user.

The threat can be represented pictorially as Fig. 4.

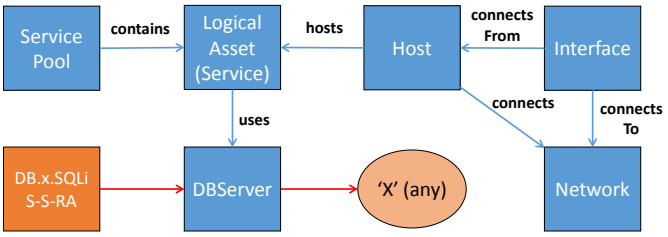

Fig. 4. Semantic threat model for DB.X.SQLIS-S-RA

Next, we consider what control options could be used to counteract it. These too can be represented in the form of diagrams, showing which type of control(s) are needed associated with which assets. The first option, which applies to any threat that works by exploiting a software vulnerability, is to patch the vulnerable asset as shown in Fig. 5. Of course, the SOFTWAREPATCHING control is only available once a patch is available to fix the vulnerability in the logical asset. There may also be some delay in applying the patch (Cyber Essentials recommends security patches should be applied automatically or within 14 days).

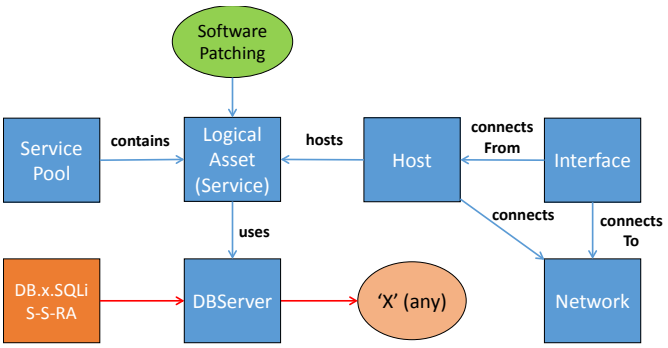

Fig. 5. Software patching against threats of type DB.X.SQLIS-S-RA

An alternative approach might be to restrict access to the network so only fully trusted users can use the vulnerable service, as shown in Fig. 6:

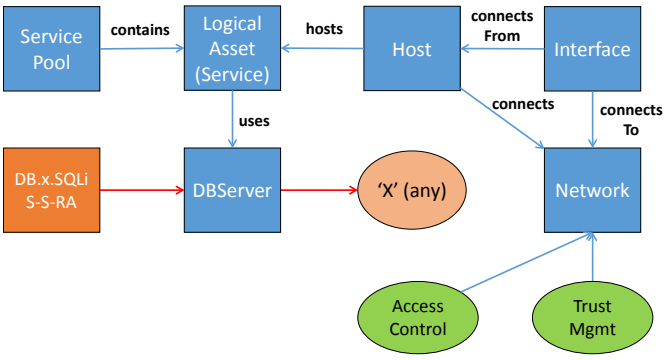

Fig. 6. Access control against threats of type DB.X.SQLIS-S-RA

Note that we cannot block this particular threat by controlling access only to the vulnerable service, because the threat can be executed by an anonymous attacker. One must block malicious requests before they reach the service over the network. Another way to do this might be to use a firewall rule to prevent traffic from reaching the HOST from that network:

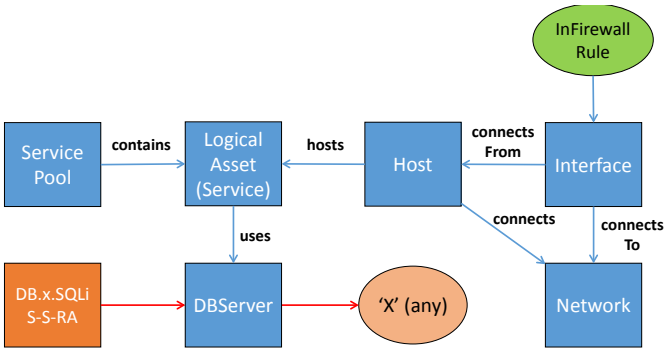

Fig. 7. Firewalling against threats of type DB.x.SQLiS-S-RA

This is usually not a viable strategy because it would also block legitimate (and trusted) users of the service. However, it is sometimes useful where the Host is connected to multiple networks and legitimate users can avoid using this particular network.

E. Mapping to CVE/CWE classification

Threats similar to the above were derived from the CVE database, based on the analysis of typical examples with significant threat severity. One interesting finding is that the CVE/CWE characterisation is actually not very well matched to the semantic classification based on the involved system assets. In retrospect this is not very surprising, since CVE entries describe software vulnerabilities and typically focus on specific code modules or even individual lines of code.

In practice, this is revealed by the fact that most OPTET threat classes correspond to multiple CWE categories. The CWE categories distinguish the nature of the vulnerability, e.g. whether it involves a buffer overflow or a failure to sanitise input, etc. However, many different CWE cases can be exploited in the same way, i.e. by using the same pattern of asset interactions to gain access to the vulnerable asset, and in most cases producing the same potential consequences. Equally, many CWE cases can be exploited in multiple contexts. Hence we find a many-to-many mapping between CWE categories and semantic threat classes. The only exceptions are CWE classes related to very specific types of exploits, notably SQL injection and XSS attacks. These map onto a small number of threat classes, due to the fact that the corresponding exploits imply specific configurations of the involved assets.

TABLE IV. SEMANTIC THREAT CLASSES RELATED TO CWE CLASSES

Threat Class	Description	Related CWE classes
H.x.SoH-H-LA	Exploitation of a local bug in a Host by an unauthenticated physically local user. Also other effects of local bugs.	20, 74, 94 (input errors) 119 (buffer overflow) 134 (string format issue) 287 (authentication issue) 362 (race condition)
H.x.SoH-H-LU	Exploitation of a local bug in a Host by an authenticated local user, e.g. to gain privileges	20, 74, 94 (input errors) 119 (buffer overflow) 134 (string format issue) 200 (info leakage) 284 (authorization issue) 287 (authentication issue) 362 (race condition)
L.x.SoLA-L-LU	Exploitation of bugs in a LogicalAsset by an authenticated local user Also any local crash of the LogicalAsset	As above
H.x.HoLA-S-LU	Exploitation of a bug in a logical asset to compromise the Host	20, 74, 94 (input errors) 22, 59 (file path/link errors) 77, 78 (command injection) 119 (buffer overflow) 134 (string format issue) 200 (info leakage) 284 (authorization issue) 287 (authentication issue) 362 (race condition)
H.x.RoH-H-RA	Exploitation of a Host bug by a remote anonymous user	20, 74, 94 (input errors) 119 (buffer overflow) 134 (string format issue) 200 (info leakage) 284 (authorization issue) 287 (authentication issue) 345 (data authenticity issue) 362 (race condition)
H.x.RoH-H-RU	Exploitation of a Host bug by a remote authenticated user	None. Remote authenticated users must enter via a logical asset.
H.x.RoS-H-RA	Exploitation of a host bug by a remote anonymous service user to compromise the Host	20, 74, 94 (input errors) 22, 59 (file path/link errors) 77, 78 (command injection) 119 (buffer overflow) 134 (string format issue) 200 (info leakage) 284 (authorization issue) 287 (authentication issue) 345 (data authenticity issue) 362 (race condition)
H.x.RoS-H-RU	Exploitation of a host bug by a remote authenticated service user to compromise the Host	As above
H.x.RoS-S-RA	Exploitation of a service bug by a remote anonymous service user to compromise the Host	As above
H.x.RoS-S-RU	Exploitation of a service bug by a remote authenticated service user to compromise the Host	As above
S.x.RoS-S-RA	Exploitation of a service bug by a remote anonymous service user to compromise the Service	As above
S.x.RoS-S-RU	Exploitation of a service bug by a remote authenticated service user to compromise the Service	As above
H.x.RoC-H-RA	Exploitation of a host bug by a remote anonymous attacker via some hosted client	As above
H.x.RoC-C-RA	Exploitation of a client bug by a remote anonymous attacker to compromise its Host	As above

Threat Class	Description	Related CWE classes
C.x.RoC-C-RA	Exploitation of a client bug by a remote anonymous attacker to compromise the client	As above
DB.x.SQLiS-S-RA	Exploitation of a service bug by a remote anonymous attacker to compromise a back end DB	89 (SQL injection)
DB.x.SQLiS-S-RU	Exploitation of a service bug by a remote authenticated attacker to compromise a back end DB	89 (SQL injection)
C.x.XSSa-W-RA	Exploitation of a website bug to attack a client by persuading them to click on a malicious link resulting in an XSS attack.	79 (Cross site scripting)
C.x.XSSb-W-RA	Same as C.x.XSSa-W-RA except that the attacker uploads the malicious content to the vulnerable website, where it is stored and later served to clients.	79 (Cross site scripting)
C.x.Phish-W-RA	Same as C.x.XSSa-W-RA except that the client is sent the malicious link via a Phishing email.	79 (Cross site scripting)

The obvious conclusion from this classification exercise is that CVE/CWE classification is not especially useful if one wants to relate vulnerabilities and the associated threats to the overall design or configuration of a system. Indeed, the CVE/CWE classification is primarily aimed at the software developer, who needs to know about the vulnerability so they can produce a fix, and in that it is hugely successful. However, it is not very helpful for a system designer attempting to use a security-by-design approach. The designer needs to know about exploitable patterns of assets and asset relationships so they can minimise exposure to threats. This knowledge is not readily obtainable from the CVE/CWE classification, but it is provided by the semantic threat modelling approach.

F. *Other common threats*

While our main concern in this study was to assess common threats arising from known vulnerabilities, it makes sense to include some additional threat classes corresponding to misconfiguration or other weaknesses including user error. We therefore retained other threat models from the original ontology corresponding to:

- message snooping on a network between a CLIENT and Service seeking to breach confidentiality of either side;
- impersonation of a CLIENT to a SERVICE or vice versa, seeking to damage either side;
- failure to securely configure a HOST, allowing remote penetration of an open service;
- failure to securely configure a HOST, allowing a local attacker to gain control by introducing their own software, e.g. on a USB storage device;
- failure to restrict access to a private network.

The last entry in TABLE IV. is also strictly an additional threat class, because the CVE data does not consider how a client is persuaded to initiate the interaction with a vulnerable website that leads to an XSS attack.

V. SME NETWORK MODELLING AND ANALYSIS

A. *Network Model 1: A Traditional SME Network*

The first exemplar SME network used in the project was based on an interview with a micro-SME (actually a sole trader) that is well known to one of the project team. This particular SME works with customers almost entirely by digital means. However, it was established 15 years ago before the advent of cloud services. Consequently the IT network is along 'traditional' lines often found in larger organisations, with a well-defined perimeter protected by a firewall, and servers running essential services in house.

The SME is actually a sole trader working from a small office home office (SOHO) base. The work is carried out there, so this SME has no 'mobile' or 'remote' workers, and their entire network is located in house behind a boundary firewall provided by their broad band router. The router allows external access to one server that runs their website and their email server, plus a webmail interface providing user access to email from within the office and during occasional (non-work related) trips away. The firewall does not allow access to the internal LAN from this server, which therefore represents a demilitarised zone (DMZ). The sole trader does make use of external services, including customer services for secure file sharing, and 3rd party websites used to research topics related to the customer's projects. System administration is handled by a second person who does the work as a favour. They have considerable IT expertise, but the work is done in their spare time so the configuration is kept quite simple using Linux servers running minimal software and managed via command line tools. The system administrator is aware of Cyber Essentials, and uses it as a guide, but the SME is not Cyber Essentials certified. The diagram of this network produced at the end of the interview is shown in Fig. 8.

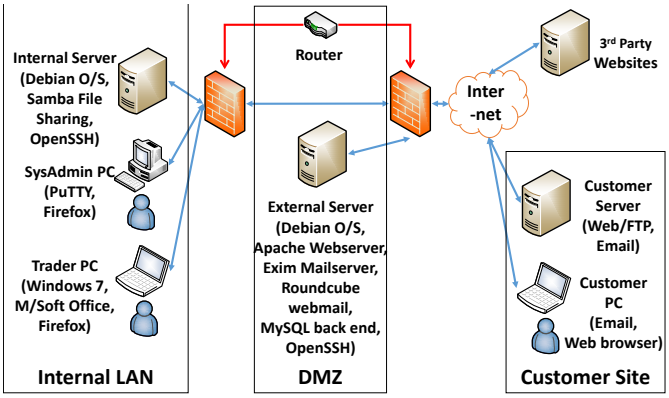

Fig. 8. SME network: conventional representation

To map analyse this network using OPTET models of threats and security controls, we must first create a semantic representation as shown in Fig. 9. Note that the INTERFACE and

SERVICEPOOL asset subclasses are not shown in this diagram, as their presence can be automatically inferred from the structure of the generic model given the asset classes that are shown.

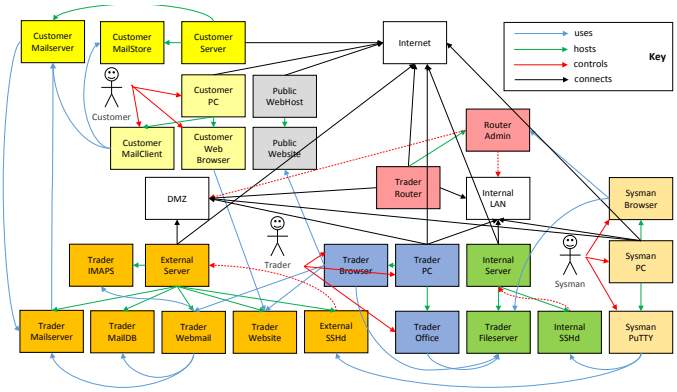

Fig. 9. SME Network Model 1: OPTET representation

The colour coding in this model indicates the HOSTS with which each asset is associated. We have the following assets:

- The SME's broadband router, which is administered through a web based admin interface and provides connectivity to and from the INTERNET and between the two LAN partitions within the SME's own network.
- The SME systems administrator's Windows PC, running a web browser and an SSH client used to control the router and (via SSH connections) the two servers;
- The trader's Windows PC, running Office software which is used to create and process files for customers, and a browser used to access email (via a webmail interface), and 3rd party websites used for research;
- The SME's internal Linux file server which is on the internal LAN and provides file and print sharing, plus SSH allowing access by the SYSMAN;
- The SME's external Linux file server which is in the DMZ and runs the SME's mail server, IMAP store and webmail interface (with a SQL database back end) and the SME's external website, plus SSH allowing access by the SYSMAN;

The diagram also shows some assets that lie outside the SME's own network, and are not under the SME's control:

- A sample customer's server which runs their own email system and remote file access services;
- A sample customer's workstation which runs their email and web clients;
- A sample 3rd party running a server hosting a public website.

These assets are included because they may be involved in threats to the SME's own assets, e.g. if a public website fails to authenticate access by a Trader, then someone else may be able to impersonate the SME. It is also possible that the SME could compromise those external assets, e.g. if the SME's own website is vulnerable to XSS attacks, then their clients could be compromised as a result.

Note that the semantic model is strictly a set of asset classes and their relationships, so this model could also be applied to a larger SME in which there may be multiple traders and even system managers with their associated workstations and software, each of which would be represented by instances of the corresponding classes.

The network model was then 'compiled' using automated semantic reasoning methods, mapping generic threat models such as the one from Fig. 4 onto the topology of this particular network. The result of that particular mapping is shown in Fig. 10.

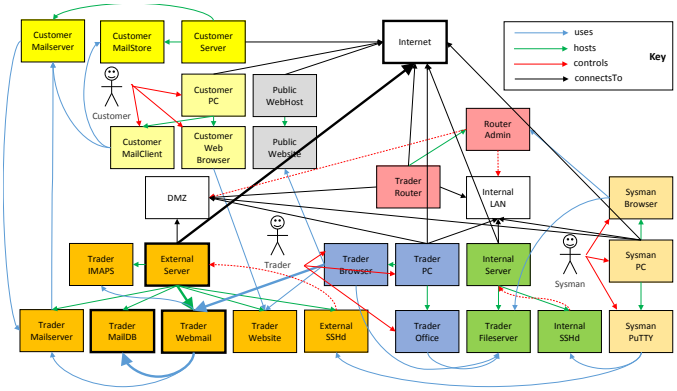

Fig. 10. Threat pattern located in the system model

Note that only four of the six generic asset classes from Fig. 4 are matched to the system-specific assets in Fig. 10. This is because the INTERFACE and SERVICEPOOL subclasses are not shown explicitly in Fig. 10. The threat identified by this mapping process is a SQL injection attack launched from the INTERNET against the trader's webmail service, compromising the back end database which holds the user email addresses and stored mail.

This automated threat mapping reduces the effort of threat identification as well as the risk of human error when systems become more complex. It also produces a well-defined model in which the effect of controls can be automatically inferred by reference back to generic control models like those in Fig. 5 to Fig. 7. This makes it much easier to identify threats and find ways to address them, or (in the context of this study) to determine the effect of a given set of controls in terms of the number of threats they can address.

One drawback of the semantic modelling approach is that having included some customer and 3rd party assets, the automated reasoning tools will treat them as part of the system and generate a complete set of threats for them. While some of those threats affect the SME's system or are consequences of the SME's own vulnerabilities, others are entirely independent of the SME. Our analysis of Cyber Essentials should exclude these totally independent threats before or after considering the effective of Cyber Essentials controls. Technically, they are acceptable residual threats, i.e. it is acceptable (to the SME) if they are not addressed by the Cyber Essentials control strategy.

The semantic analysis also generates threats which are technically valid and do affect the SME's assets, but which can be discounted for practical reasons. For example, one of the generic threats covers snooping of messages by an interloper on

a network connecting the HOSTS of two interacting LOGICALASSETS. In practice, this traffic could pass through any network connecting the two Hosts where these LOGICALASSETS are located. The semantic mapping procedure will generate two snooping threats for data exchanged between the TRADERFILESERVER and the TRADEROFFICE assets: one for snooping on the INTERNALLAN, and one for snooping on the Internet. In practice, of course, the second of these threats is of little concern because the data would not normally be routed via the Internet.

The third type of acceptable residual threat is one that does not need to be addressed because the damage caused is negligible. A good example might be snooping traffic between the SME's TRADERWEBSITE and its CUSTOMERWEBBROWSER clients. Since the TRADERWEBSITE is available to anyone connected to the Internet, it matters little if its content is inspected on its way to a CUSTOMER (the corruption of that content would be a different matter, of course).

For this model, a total of 617 threats were identified that may affect the system as shown in Fig. 9. A manual inspection of these threats led to 124 being tagged as one of the types of acceptable residual threats as described above. This left 493 relevant threats to be addressed by Cyber Essentials.

To check the effectiveness of Cyber Essentials, control elements were added control elements to the model from Fig. 9. This was done one Cyber Essentials chapter at a time, giving an idea of how each chapter contributes to the security of the system. The threats were then classified using the control rules as encoded in the generic threat model described in Section IV, so that some threats were classified as having been addressed by the specified controls. The number of threats that were still not addressed gives an indication of the residual exposure.

The addition of controls to the network model obviously has to take account of what is appropriate, e.g. implementing boundary firewalls should not block traffic to the trader's external website. The control elements introduced were as follows (see Section IV.C for the definition of controls):

- Chapter 1: Boundary Firewalls: SECURECONFIGURATION of the TRADERROUTER, ACCESSCONTROL plus TRUSTMGMT (only fully trusted users) for the TRADERROUTER and ROUTERADMIN (the LOGICALASSET that implements firewall rules), INFIREWALLALL between each HOST and the INTERNET, and between each HOST and the DMZ network, except for the EXTERNALSERVER which must be accessible from both so internal and external users can reach the SME's website and email servers. Note that this chapter focuses on perimeter firewall protection, but also mandates secure configuration and access control for the routing devices and software.
- Chapter 2: Secure Configuration: SECURECONFIGURATION for all the SME's other HOSTS; INFIREWALLALL between each PC host and the INTERNALLAN (i.e. a personal firewall blocking inbound connections to these PCs even from the LAN).
- Chapter 3: User Access Control: ACCESSCONTROL and TRUSTMGMT for the DMZ and INTERNALLAN, and also for the INTERNALSSHD and EXTERNALSSHD services used to manage the internal and DMZ servers; ACCESSCONTROL also used for the two servers and the TRADERWEBMAIL, TRADERIMAPS and TRADERMAILDB services.
- Chapter 4: Malware Protection: ANTIMALWARE controls introduced at all the SME's Hosts.
- Chapter 5: Software Patch Management: SOFTWAREPATCHING introduced for all the SME's Hosts and LOGICALASSETS.

Introducing these measures one section at a time, then automatically checking the status of each system specific threat class against the generic threat control rules yielded the following results:

TABLE V. CYBER ESSENTIALS EFFECTIVENESS: TRADITIONAL SME

Chapter	#Relevant Threat Classes	#Addressed	%Effect-iveness
None	493	0	0%
Boundary Firewalls	493	65	13%
Plus Secure Configuration	493	75	15%
Plus User Access Control	493	177	36%
Plus Malware Protection	493	262	53%
Plus Software Patch Mgmt	493	390	79%

The overall effectiveness score from TABLE V. is calculated as the proportion of unacceptable threats that are addressed by the specified controls. Software patch management made the biggest single contribution, addressing 128 identified threats excluding any that were already addressed by other Cyber Essentials protection. This is not surprising, given that the threat ontology was based mainly on software vulnerability exploits from the CVE database, and prompt patching should prevent most of these known exploits.

Cyber Essentials claims to cover 80% of 'common' cyber security threats, and our analysis suggests this claim is broadly valid for a traditional SME with a well-defined perimeter. The threats that are not addressed include impersonation attacks, network snooping attacks and XSS attacks involving 3rd party websites (whose security is not guaranteed). The threats to message security can be addressed using service authentication and encryption, and the XSS threats by mail scanning. However, in both cases the protection afforded would be more effective if users are also trained in safe use of the Internet.

B. Network Model 2: A Cloud Based SME network

The second network model represents a more modern SME start up, in which basic public services like the company website and email are rented in the cloud. The two commonly used paradigms are:

- Infrastructure-as-a-Service (IaaS): in which the SME rents server capacity, provisioned as one or more virtual machines, which are managed by the SME;
- Software-as-a-Service (SaaS): in which the SME rents a turnkey solution providing the services they need, managed by the cloud service provider.

The SMEs we interviewed showed a clear preference for SaaS. One of its selling points is the fact that the cloud service provider handles maintenance of the underlying servers, and also the software they run to provide the services. This includes security measures such as secure configuration of the data centre, applying security patches to the operating systems and application software, etc. The SME still has to do some management, e.g. of user accounts, but far less than if they ran their own server in-house or on a cloud-hosted VM. This is likely to be attractive to small organisations that lack deep IT security expertise, i.e. the type of SME for which Cyber Essentials was developed. The SaaS paradigm was therefore used in the example network model.

Aside from this change, Network Model 2 was kept as similar as possible to Network Model 1 to simplify direct comparisons between the results. The customer and 3rd party assets are unchanged. There is still an INTERNALLAN with a router providing access to the Internet, but no DMZ since the services that ran there in Model 1 are now in the cloud and accessed via the Internet. The Traders still use office software and have an internal server for file and print sharing. The system manager still has a separate PC from which to manage the network, but web and email services are now managed via a web CLOUDADMIN interface. With only one internal server to manage, the SYSMAN does this directly by logging into the console rather than using remote access via PuTTY and SSHd.

The resulting network is shown in Fig. 11. Note that the CLOUDHOST represents a slice of the cloud data centre used to run the SaaS services shown. This does not correspond to an IaaS VM, as the SME has no access to the data centre at that level. The CLOUDADMIN interface allows them to specify their users, configure the look and feel of their web presence, maintain the content of their website, etc.

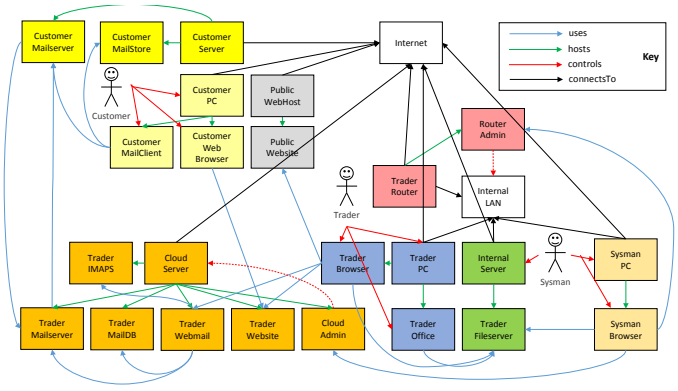

Fig. 11. SME Network Model 2: Cloud Based SME

Compiling this model produced a list of 408 threat classes of which 108 were considered to be acceptable residual threats. Thus 300 threat classes were considered relevant to the study of Cyber Essentials effectiveness.

In this model, prior to applying any Cyber Essentials measures of their own, the SME will benefit from some security implemented by the cloud service provider. In particular, we assume the cloud service provider has SECURECONFIGURATION and ANTIMALWARE protection for the CLOUDHOST, and SOFTWAREPATCHING for the CLOUDHOST and all the LOGICALASSETS hosted there. Thus some protection is assumed to be in place before the SME implements any Cyber Essentials measures of their own.

The Cyber Essential chapters were then added one at a time as before:

- Chapter 1: Boundary Firewalls: SECURECONFIGURATION at the TRADERROUTER, ACCESSCONTROL plus TRUSTMGMT (only fully trusted users) for the TRADERROUTER and ROUTERADMIN, and INFIREWALLALL between each HOST and the INTERNET, there being no EXTERNALSERVER and no DMZ to worry about.
- Chapter 2: Secure Configuration: SECURECONFIGURATION for the SME's other HOSTS, and INFIREWALLALL between each PC host and the INTERNALLAN.
- Chapter 3: User Access Control: ACCESSCONTROL and TRUSTMGMT for the INTERNALLAN and CLOUDADMIN (which replaced the SSHd remote management); and ACCESSCONTROL also for the internal server and the TRADERWEBMAIL, TRADERIMAPS and TRADERMAILDB services.
- Chapter 4: Malware Protection: ANTIMALWARE introduced on all the SME's in-house HOSTS.
- Chapter 5: Patch Management: SOFTWAREPATCHING added for all the SME's in-house HOSTS and LOGICALASSETS.

Introducing these measures one section at a time, then automatically checking the status of each system specific threat class against the generic threat control rules yielded the following results:

TABLE VI. CYBER ESSENTIALS EFFECTIVENESS: CLOUD BASED SME

Chapter	**#Relevant Threat Classes**	**#Addressed**	**%Effect-iveness**
None	300	0	0%
Cloud Service Provider Security	300	91	30%
Plus Boundary Firewalls	300	128	43%
Plus Secure Configuration	300	137	46%
Plus User Access Control	300	143	48%
Plus Malware Protection	300	160	53%
Plus Software Patch Mgmt	300	217	72%

The first striking point in this data is that the total number of threats is lower than before. This is mainly due to the fact that there is no DMZ, so whereas before there would have been separate threat classes involving attacks within the DMZ, now we only have classes for the Internet-related threats.

The next noteworthy point is that the security provided by the cloud service provider has considerable value, addressing 30% of the significant threats. Many though not all threats to the SME's web and email services are covered by this. However, it must be noted that not all cloud service providers are equally helpful. It is *essential* for security that the SME fully understands what controls are provided by the cloud service, and ideally that this correspond to the assumptions we have made. This point is *not* really addressed in Cyber Essentials, and would be a useful extension of the guidance provided.

Observe however that the SME's own efforts to implement Cyber Essentials here are *less* effective than in the previous case. This is because nothing the SME does will provide much additional protection to their cloud-hosted services, or to communications via the Internet with those services. Boundary firewalls, malware protection and software patching still have value in protecting in-house resources, and the total number of unaddressed threats is actually *lower* than before, but the proportion of unacceptable threats addressed is also lower at *only* 72%, well below the claimed effectiveness level.

C. SME Network Model 3

The last model represents an SME whose workers spend most of their time at remote locations, either working from home or at customer sites. This involves a more extreme de-perimeterisation of the SME network. To handle this, the SME must expose additional services to the Internet, e.g. giving the Traders access to a shared file system so they can exchange documents and data.

In our example network model, we assume the SME takes the plunge and procures a complete IT infrastructure from a cloud service provider such as Huddle. This includes 'office' applications needed by the traders, e.g. using something like Microsoft's Office 360 if they really are office applications. Similar SaaS offerings are now available for other applications including CAD and computer aided engineering (CAE) analysis, as well as customer relationship management, HR and payroll systems, etc.

The resulting model is shown in Fig. 12:

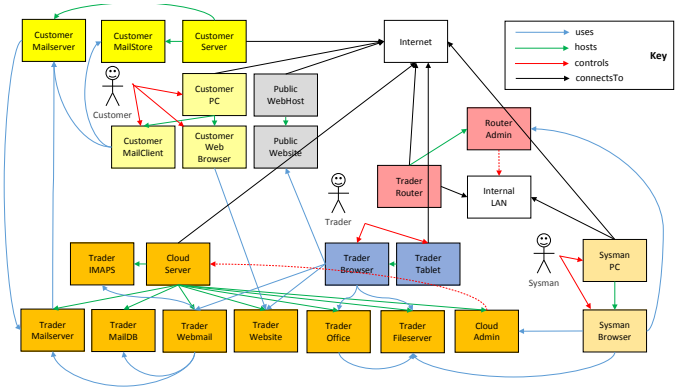

Fig. 12. SME Network Model 3: SME with Mobile Workers

Note that the INTERNALSERVER has gone, as practically all the SME's core services are now hosted in the Cloud. The TRADERPC is now replaced by a TRADERTABLET with 3G/4G as well as WiFi connectivity so it can connect via the INTERNET from customer sites or even on the road. These devices may include some '*Bring Your Own Device*' (BYOD), i.e. Traders using their own personal devices rather than being supplied with devices configured by their SYSMAN. The only assets that are still behind a traditional network perimeter are the SYSMAN's workstation and web browser, through which they administer the company's services in the Cloud.

Compiling this model produced a list of 411 threat classes, of which 104 were considered acceptable residual threats. Thus 307 threat classes are relevant to the study of Cyber Essentials effectiveness.

As with Model 2, the SME can expect some protection from the security implemented by the cloud service provider. The same assumptions have been made, i.e. that the cloud service provider has SECURECONFIGURATION and ANTIMALWARE protection for the CLOUDHOST, and SOFTWAREPATCHING for the CLOUDHOST and all the LOGICALASSETS hosted there, which now include the TRADEROFFICE and TRADERFILESERVER services.

The Cyber Essentials chapters are then modelled via:

- Chapter 1: Boundary Firewalls: SECURECONFIGURATION at the TRADERROUTER, ACCESSCONTROL plus TRUSTMGMT (only fully trusted users) for the TRADERROUTER and ROUTERADMIN, and INFIREWALLALL between the SYSMANPC and the INTERNET, and between the TRADERROUTER and the INTERNET (i.e. isolating the INTERNALLAN, for what it is worth).
- Chapter 2: Secure Configuration: SECURECONFIGURATION for all the SME's other HOSTS (the SYSMANPC and TRADERTABLET hosts), INFIREWALLALL between the SYSMANPC and the INTERNALLAN, and INFIREWALLALL between the TRADERTABLET and the INTERNET.
- Chapter 3: User Access Control: ACCESSCONTROL and TRUSTMGMT for the INTERNALLAN and CLOUDADMIN (which replaced the SSHd remote management), ACCESSCONTROL also for the TRADERTABLETS, SYSMANPC and the various cloud based services.
- Chapter 4: Malware Protection: ANTIMALWARE introduced on all the SME's Hosts.
- Chapter 5: Patch Management: SOFTWAREPATCHING added for the SYSMANPC and TRADERTABLET hosts and web browsers.

Note that ANTIMALWARE and PATCHMANAGEMENT might not be reliable for user-managed, BYOD devices. However, for the purposes of this study it was assumed that the SME could impose these controls, e.g. by inspection of user devices before they are used for work purposes.

Introducing these measures one section at a time, then automatically checking the status of each system specific threat class against the generic threat control rules yielded the following results:

TABLE VII. CYBER ESSENTIALS EFFECTIVENESS: CLOUD BASED SME

Chapter	#Relevant Threat Classes	#Addressed	%Effect-iveness
None	307	0	0%
Cloud Service Provider Security	307	130	42%
Plus Boundary Firewalls	307	150	49%
Plus Secure Configuration	307	158	51%
Plus User Access Control	307	164	53%
Plus Malware Protection	307	189	62%
Plus Software Patch Mgmt	307	213	69%

The first point worth noting is that while the total number of relevant threats is still *lower* than in Model 1 (due to there being *fewer* networks over which attacks could in principle be made), we have *more* threats than in Model 2. This comes mainly from the fact that we have fewer acceptable residual threats, because with most SME assets now outside the perimeter very few communications follow 'safe' paths.

The effect of security measures assumed to be implemented by the cloud service provider is proportionately greater, due to the fact that all the SME's core services are now hosted in the cloud. But despite this, Cyber Essentials is less effective than in Model 2, as adding more security measures within the SME becomes a case of diminishing returns.

In this case, we examined the nature of the threats not being covered. The largest category turned out to be confidentiality breaches when communicating over the public Internet. These threats arise more often in Model 3 because all services are accessed via the Internet, so none of the snooping and spoofing threats can be considered acceptable due to the communications being internal to the SME. Rigorously applying SecureTransport with ServiceAuthentication would address many of these threats – indeed adding these controls to the relevant assets increased overall coverage to 84%.

VI. CONCLUSIONS AND FUTURE WORK

The objective of this work is to provide an analytical study to investigate the effectiveness of the five Cyber Essentials controls in the context of typical SME networks. The approach, based on system modelling and automated threat identification and analysis has proven very effective in allowing the effect of Cyber Essentials control strategies to be determined.

Our models created and used were based on insights gained from published surveys backed up by interviews with SMEs. The three models used spanned a range of typical network configurations: SMEs with in-house IT systems and well-defined network boundaries, SMEs that make use of cloud services, and SMEs with mobile users connecting from remote locations, possibly using their own (personal) devices.

Our analysis and interviews also suggested that cyber security remains a low priority for many small businesses. Staff are typically unaware of cyber security risks, and even IT support staff are not aware of schemes like Cyber Essentials, even where they are aware of (and address) the need for cyber security.

Our findings suggest that Cyber Essentials would live up to its claim to address about 80% of common threats for SMEs with traditional in-house IT networks. However, it falls short of this for SMEs that make significant use of cloud services, or who have mobile workers even if they use company owned and configured devices. Our analysis suggests that the following additions to Cyber Essentials would have significant value:

- guidelines for the selection and use of cloud services, including what security mechanisms should be handled by the service provider;
- guidelines for communication security, including rigorous use of service authentication and encrypted communications;
- extension of malware protection to include the use of mail scanning to detect and block a significant proportion of XSS and phishing attacks;
- guidelines on the training of users in the safe use of the Internet, including the effective use of communication security and how to handle suspicious email.

To address the lack of awareness of cyber security risks, we recommend further research on how best to communicate about threats and countermeasures such as Cyber Essentials to IT support staff. One approach would be to make semantic threat analysis tools and knowledge base accessible to SMEs. For example, it may be helpful to provide SMEs with an online service allowing them to describe their network via a questionnaire, then get a list of potential threats, and explore the use of Cyber Essentials or other security guidelines to counter those threats.

The threat knowledge base used in this work was based on an analysis of the well-known CVE database. The main finding from this analysis is that the CVE/CWE classification is well suited to the needs of software developers, helping them to understand vulnerabilities and fix them or (ideally) avoid introducing them in the first place. However, it is poorly suited to system designers seeking to introduce security by design at system level, or system managers who need to assemble different system components and introduce security measures to protect them.

We therefore recommend that a semantic classification approach should be used in conjunction with the CVE-style classification to describe common threats. The semantic threat classes depend not on the nature of the software vulnerability, but on the way the vulnerability relates to interdependent system assets, and on the security measures needed to counteract the threats. Establishing a database based on these principles, ideally cross-references with CVE entries would substantially increase the usefulness of the threat data collected by the community as a whole.

ACKNOWLEDGMENT

To be added in the non-anonymous version.

REFERENCES

[1] BIS Research Paper 214, March 2015.

[2] Cyber Essentials Scheme: Requirements for basic technical protection from cyber attacks, UK Department for Business, Innovation & Skills (BIS) and the UK Cabinet Office, June 2014.

[3] Government sets the bar for Cyber Risk with Cyber Essentials, see http://www.continuityforum.org/content/news/178023/government-sets-bar-cyber-risk-cyber-essentials, June 2014.

[4] ISO/IEC 15408-1:2009: Information technology: Security techniques -- Evaluation criteria for IT security, Part 1: Introduction and general model," 2009.

[5] ISO/IEC 27001:2013, Information Technology - Security Techniques - Information Security Management Systems – Requirements, 2013.

[6] ISO/IEC 27005:2011, Information technology - Security techniques - Information security risk management, 2011.

[7] F. Swiderski and W. Snyder, Threat modelling, Microsoft Press, 2004.

[8] Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R. & Murukan, A. "Improving Web Application Security: Threats and Countermeasures," Microsoft Corporation, Redmond, USA, 2003.

[9] N. R. Mead and T. Stehney, "Security quality requirements engineering (SQUARE) methodology," ACM SIGSOFT Softw. Eng. Notes, vol. 30, no. 4, p. 1, 2005.

[10] J. Juerjens, "Using UMLsec and Goal Trees for Secure Systems Development," in ACM Symposium on Applied Computing, SAC 2002, 2002, pp. 1026–1030.

[11] Stasinopoulos, A., Ntantogian, C., & Xenakis, C. (2013, December). The weakest link on the network: Exploiting ADSL routers to perform cyber-attacks. In Signal Processing and Information Technology (ISSPIT), 2013 IEEE International Symposium on (pp. 000135-000139). IEEE.

[12] B.Schneier , Attack Trees: Modeling security threats, in : Dr. Dobbs"s Journal (1999).

[13] T. R. Ingoldsby, Attack tree-based threat risk analysis, Amenaza Technologies Ltd, 2009.

[14] D. Ramsbrock, R. Berthier, and M. Cukier, "Profiling attacker behavior following SSH compromises," 37th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks, DSN 2007, pp. 119–124, 2007.

[15] P. Xuena and Z. Hong, "An ' Attacker Centric ' Cyber Attack Behavior Analysis Technique," in Advanced Communication Technology, The 9th International Conference on, 2007, pp. 2113 – 2117.

[16] "Common Attack Pattern Enumaration and Classification." [Online]. Available: https://capec.mitre.org/

[17] Saitta, P., Larcom, B., & Eddington, M. (2005). Trike v1 Methodology. http://www.octotrike.org/papers/Trike_v1_Methodology_Document-draft.pdf.

[18] Caralli, R.A., Stevens, J.F., Young, L.R. & Wilson, W.R., Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process, Software Engineering Institute, Carnegie-Mellon, May 2007.

[19] C. Blanco, J. Lasheras, E. Fernandez-Medina, R. Valencia-Garcia and A. Toval, "Basis for an integrated security ontology according to a systematic review of existing proposals.," Comput. Stand. Interfaces , vol. 33, no. 4, pp. 372-388, 2011.

[20] S. Fenz and A. Ekelhart, "Formalizing information security knowledge," in International Symposium on Information, Computer, and Communications Security, Sydney, Australia, 2009.

[21] German Federal Office for Information Security BSI, "IT Grundschutz Manual," 2004.

[22] M. Surridge, B. Nasser, X. Chen, A. Chakravarthy and P. Melas, "Run-Time Risk Management in Adaptive ICT Systems," in Eighth International Conference on Availability, Reliability and Security (ARES), Regensburg, Germany, 2013.

[23] A. Chakravarthy, S. Wiegand, X. Chen, B. Nasser and M. Surridge, "Trustworthy Systems Design using Semantic Risk Modelling," in Procs 1st International Conference on Cyber Security for Sustainable Society, Coventry, UK, 2015.

[24] Vavoulas, N., & Xenakis, C. (2011). A Quantitative Risk Analysis Approach for Deliberate Threats. Critical Information Infrastructures Security, 13–25.

[25] Zhou, Z., & Hu, C. (2008). Study on the E-government Security Risk Management. International Journal of Computer Science and Network Security, 8(5), 208–213.

[26] "Common vulnerabilities and exposures" (2005), https://cve.mitre.org/.

[27] Available via http://www.serscis.eu/releases/serscis-ontology.zip.

[28] "Semantically Enhanced Resilient and Secure Critical Infrastructure Services" (2008-2012). See http://www.serscis.eu/?page_id=6.

[29] "OPerational Trustworthiness Enabling Technologies" (2012-2015). See http://cordis.europa.eu/project/rcn/105733_en.html.