Exploring a Controls-Based Assessment of Infrastructure Vulnerability

Oliver J. Farnan and Jason R. C. Nurse

Cyber Security Centre, University of Oxford, Oxford, UK
firstname.lastname@cs.ox.ac.uk

Abstract. Assessing the vulnerability of an enterprise's infrastructure is an important step in judging the security of its network and the trustworthiness and quality of the information that flows through it. Currently, low-level infrastructure vulnerability is often judged in an ad hoc manner, based on the criteria and experience of the assessors. While methodological approaches to assessing an organisation's vulnerability exist, they are often targeted at higher-level threats, and can fail to accurately represent risk. Our aim in this paper therefore, is to explore a novel, structured approach to assessing low-level infrastructure vulnerability. We do this by placing the emphasis on a controls-based evaluation over a vulnerability-based evaluation. This work aims to investigate a framework for the pragmatic approach that organisations currently use for assessing low-level vulnerability. Instead of attempting to find vulnerabilities in infrastructure, we instead assume the network is insecure, and measure its vulnerability based on the controls that have (and have not) been put in place. We consider different control schemes for addressing vulnerability, and show how one of them, namely the Council on Cyber Security's Top 20 Critical Security Controls, can be applied.

1 Introduction

Finding vulnerabilities is a difficult and arduous task [39]. Nevertheless, assessing the vulnerability of computer systems and networks to attacks is a critical enterprise security activity. This includes two major types of vulnerability assessment: low-level (e.g. penetration tests or automated vulnerability scans looking for vulnerabilities such as MS08-067[1]) and high-level (e.g. ISO 27001 [20] audits looking for the existence of appropriate cryptographic storage). Both are required for various types of security risk assessments, and assist in making critical business decisions, demonstrating an organisation's security credentials, and determining the risk of compromise of its information assets. In the language of risk assessment, vulnerabilities are identified and then mitigated by controls. These are taken into consideration (sometimes with additional factors such as specific threats or threat actors) when used to assess the security of the scoped infrastructure.

[1] MS08-067 is a low-level vulnerability in the Windows Server Service that allows remote code execution when sent a specially crafted RPC request.

This paper aims to explore the possibility of a more structured and rigorous approach to assessing the overall vulnerability of infrastructure against low-level vulnerabilities (e.g. whether its OpenSSL implementation is vulnerable to Heartbleed [17]). Our approach attempts to provide an unbiased assessment of infrastructure vulnerability, based on external controls from well-researched sources. We scope our work away from high-level risk analysis, where there are already several approaches in use (e.g. ISO 27001) and instead intentionally focus on the low-level technical vulnerability of an infrastructure. Moreover, at this stage we avoid higher-level factors (e.g. security procedure and policy) and do not attempt to judge the overall risk to the infrastructure.

Basing our approach on the presence or absence of controls is a different approach to previous technical assessment methodologies. Most proposed academic methods base their judgement on attempting to count the low-level vulnerabilities present, and using this as the basis for their assessment of the security of the network, infrastructure or system [1][18][21]. Similarly, within businesses, assessment of low-level vulnerability is performed by vulnerability scanning or penetration testing to discover and enumerate existing vulnerabilities on the network. Instead, our low-level evaluation takes inspiration from higher-level risk assessment methodologies, whose assessments are based on the existence (or lack thereof) of controls that an organisation has in place. An example of these is the ISO 27001:2013 standard, where auditors check to see whether a series of controls have been implemented. These controls exist at a higher-level than the vulnerabilities discovered by penetration testing or vulnerability scanning, and include checks such as whether an access control policy is in place or whether staff have been appropriately vetted.

With our approach, instead of basing the assessment on vulnerability lists and the vulnerabilities present, we base the assessment on whether certain low-level controls are in place. This is more in line with the higher-level risk assessments such as ISO 27001. Whereas typical low-level vulnerability assessments assume the network is secure unless found otherwise, we argue that the network should be considered insecure until the necessary controls (relevant to that infrastructure) have been implemented. Unknown vulnerabilities will always exist and will be impossible to defend against. Controls defending against a wide variety of attacks (such as a whitelist-based firewall, or the use of a No Execute bit against memory injection attacks) provide a greater guarantee than firefighting individual vulnerabilities. This is in line with a growing belief that networks and systems should be considered insecure by default [7][8][26].

Despite the need to assess low-level vulnerability, there is not a generally accepted way of doing so. Different methods exist for measuring the higher-level risks to an infrastructure (e.g. ISO 27001), but there is no established way of providing the low level equivalent. On the one hand there exists technical testing such as vulnerability scanning tools (e.g. Nessus [38]) and penetration testing techniques, which are widely employed but often unstructured and unsystematic. On the other hand there are several highly-structured proposed academic methods, which have not been successfully adopted into widespread use.

In the remainder of this paper, we present related work that is relevant to this discussion (Section 2), describe our approach (Section 3) and discuss an example (Section 4). We then critically reflect on our proposal (Section 5) and draw conclusions and discuss areas for future research (Section 6).

2 Related Work

Currently, there are several accepted methods for organisations to assess infrastructure vulnerability. Typically these involve some type of audit or risk assessment being carried out. These can range from comparing the infrastructure to standard procedural checklists (e.g. ISO 27000), to hands on technical audits (e.g. penetration tests). Each type of assessment takes into account different factors, considerations and controls, to give feedback on the strengths and weaknesses of the system.

One problem assessors face when evaluating network security is that the assessment may be biased towards their own knowledge and previous experiences. Left to their own devices, assessors may be prone to letting the systems they have previously worked with affect the assessment of that which they are currently assessing. To combat this, strict high-level information security assessment methodologies have been created such as ISO 27001 [20] and COBIT [19]. These aim to provide an explicit framework for assessors, to ensure that a structured process is maintained, while also reducing the risk of bias affecting the results of the assessment. Assessors are then trained to follow these frameworks and be objective in their assessment.

One of the criticisms of these approaches is that they are simplistic, and can be performed as checklists of controls. Recent research aims to address this. Bhattacharjee *et al.* [5] proposed a formal method of risk assessment that aims to take other factors into account, such as asset and vulnerability dependency. Szwed and Skrzynski [37] also consider this, with a proposal based on Fuzzy Cognitive Maps. They argue their lightweight risk assessment methodology is easy to apply, and more appropriately takes into consideration the value of assets.

While widely accepted assessment methodologies are in place for the procedural side of information security, there is nothing analogous when it comes to assessing technical vulnerabilities. Technical assessors and penetration testers frequently rely on the experience of their previous assessments to judge the overall vulnerability of the network that they are testing. Technical assessments of this kind focus on specific low-level vulnerabilities (e.g. MS08-067 present on a specific host) and do not have a structured way of building on these to provide an overall assessment of the vulnerability of an infrastructure. Executive summaries are often included to provide this bigger-picture assessment, but as there is no accepted process it is often tainted by the testers' prior experience [41].

Modern penetration testing arguably began with Karger and Schell's evaluation of Multics for the US Air Force [22][23]. They performed a thorough evaluation of Multics, and found several ways to bypass its multi-level user control system. Penetration testing is at times a controversial approach, and Valli *et al.* [41]

recently assessed some of the weaknesses of relying on penetration testing. They argue that penetration testing is often not the best process to base security decisions on, and can be driven by ulterior motives. In contrast, Shah and Mehtre [33] provide an overview of current penetration testing techniques, and describe how they can be beneficial for an organisation.

Ken Thompson's 'Reflections on Trusting Trust' [39] is the classic paper on the difficulty of detecting vulnerabilities. He demonstrates the possibility of a vulnerability invisible even to thorough source code review, and gives a clever example of a vulnerability that would be difficult to detect via automated means. Nevertheless, vulnerability scanners are a common tool used for the detection of low-level vulnerabilities in systems and networks. Examples of scanners include Nessus [38], OpenVAS [29] and Core Impact [9], which are used in many proposed academic methods for assessing infrastructure vulnerability [18][21].

In practice, both vulnerability scanning and penetration testing work by targeting specific systems and devices, and sequentially testing for the existence of vulnerabilities, generally by attempting to exploit them with non-malicious payloads. Details of these vulnerabilities can be found in comprehensive databases such as CVE Details [14] or NIST NVD [28], where there is also information available on their impact (via CVSS scores). Vulnerability scanning involves scanning the infrastructure with automated tools which are designed to find and fingerprint vulnerabilities, while penetration testing involves performing the process manually, resulting in a more thorough analysis. If the infrastructure proves vulnerable to an attack it is recorded and presented in a report. A report is then produced listing the vulnerabilities that were discovered on the host.

There has also been a noteworthy amount of academic research on assessing infrastructure vulnerability. These primarily aim to judge the vulnerability of infrastructure based on the low-level vulnerabilities detected. Jajodia and Noel [21] have proposed a method for assessing the vulnerability of network topologies based on the accumulative vulnerabilities of paths into and out of the network. These vulnerabilities are detected using scanners such as Nessus [38], and can be used with the network intrusion detection system Snort [34] to correlate received alerts. Ahmed *et al.* [1] describe a method for assessing the vulnerability of a network based partly on the vulnerabilities that have historically been present. They find that if a service has a history of vulnerabilities, there is a higher probability that the service will be vulnerable in the future.

Holm *et al.* [18] analyse the effectiveness of different rubrics for judging systems vulnerabilities through CVSS scores. They base their work on the time taken to compromise systems in the cyber-defence exercise Baltic Shield [16], an exercise which pitched a red team of attackers against blue teams of defenders trying to prevent the compromise of a network. The known vulnerabilities of the systems had their CVSS scores combined using several methodologies proposed by other researchers, and the assessment of these methodologies was compared to the actual time taken to compromise the systems. The study found that simple methodologies only looking at the most serious vulnerabilities present in each system or service (based on the security belief of 'weakest link in the chain')

were not as effective at estimating the difficulty to compromise the network as those that took more information into account. Teodor *et al.* [35] followed on from this by presenting a modelling language for analysing the security of enterprise system architectures. They found that analyses using their model can be as accurate as assessments performed by security professionals. In a similar vein, Feng *et al.* [15] recently proposed a method to consider the relationship between risks. Their approach uses Bayesian networks to consider not just the vulnerability, but its context within an infrastructure.

To summarise, most of the research into proposed methods of low-level vulnerability analysis are based on the presence of vulnerabilities. Despite these proposals, none of these methodologies appear have gone on to widespread use on live systems. This is in contrast to the higher-level vulnerability assessment methodologies (e.g. ISO 27001), which are primarily controls-based. While there are many possible reasons for this, one major factor may be the difficulty in determining which low-level vulnerabilities are present. While processes that rely on knowing which low-level vulnerabilities exist may work in controlled tests, they are likely to be more difficult to implement when analysing live systems.

Not content with the frequently ad hoc nature of low-level information security defences, there have been several attempts to produce standardised lists of controls. Two examples relevant to this discussion are the Council on Cyber Security's Top 20 Critical Security Controls [12][10] and Australian Signals Directorate (ASD) 35 Cyber Security Mitigation Strategies [3].

The Council on Cyber Security's Top 20 Critical Security Controls (CSC 20) is a set of 20 security controls based on an observed need for a standardised controls list of this type. It was drawn up by an array of 'companies, government agencies, institutions, and individuals from every part of the [security] ecosystem' [10], based on an 'offence informs defence' approach. They have been widely adopted by other organisations [30], including SANS [31] and the UK Centre for the Protection of National Infrastructure (CPNI) [13]. The CSC 20 is regularly updated, and is on version 5.1 since its inception in 2012. An example of a CSC 20 control is: Limitation and control of network ports, protocols and services.

The ASD's 35 Cyber Security Mitigation Strategies is a series of 35 controls ranked in order of overall effectiveness at network protection. It places particular emphasis on their top 4 mitigation strategies, which they argue stop 85% of targeted cyber intrusions [3]. It is written at a slightly lower-level of abstraction than the CSC 20, and largely include similar and overlapping controls [11]. An example of an ASD control is: Restrict access to Server Message Block (SMB) and NetBIOS services (this example would be covered by the slightly-higher level CSC 20 control mentioned above – i.e. Limitation and control of network ports, protocols and services).

3 A Controls-Based Approach

Our approach to vulnerability assessment is based on the thesis that all infrastructures are vulnerable, and that this vulnerability can only be mitigated

with the implementation of certain controls. This is similar to higher-level risk assessment methodologies (e.g. ISO 27001). Using control schemes such as the CSC 20, we are instead exploring the possibility of basing low-level vulnerability assessments on the controls present. A system that correctly and securely implements all controls is considered as secure as it can be under the scheme, while a system that implements no controls is assessed as being insecure. This view is supported by the growing belief that we cannot take the trustworthiness of network infrastructure for granted [7][8]. Unfortunately devices are still often not built with security in mind, and contain numerous undiscovered vulnerabilities [32]. Instead of assuming technology is secure until proven otherwise, it is prudent to consider it insecure until proven secure.

Instead of attempting to create an exhaustive list of vulnerabilities or controls that affect infrastructure vulnerability, our approach relies on lists of controls compiled from other sources. We see two primary advantages in doing so. Firstly, the method itself can be static and not constantly changing. This makes the process more robust, not dependent on specific vulnerabilities, and not impacted by the changing vulnerability landscape. Secondly, there is already a large amount of research into the most effective controls to address vulnerability. Instead of replicating this work, we can build on it. Moreover, determining the vulnerabilities or controls which have the most effect on infrastructure vulnerability is a large and complex task, and far beyond the scope of this current paper.

Our approach follows a multi-step process. We envisage that the steps can be repeated using different control schemes, without losing the overall structure of the assessment. The approach is defined as follows:

Step 1 – Scope the infrastructure to be evaluated. When performing any security assessment it is important to determine exactly which assets are to be covered. This will allow the assessors focus on the area where risk has been identified, and not spend time assessing assets not considered vital to the organisation. This can be a complex process, and must consider the interdependencies within and across the organisation.

Step 2 – Select the control list to be applied. Compliance with different control lists will offer different levels of assurance to the infrastructure. Two control lists that currently meet these requirements are the CSC 20 and ASDs 35 Cyber Security Mitigation Strategies. Control schemes should be chosen with consideration of the following criteria:

- *It is appropriate for the infrastructure* – e.g. the controls it contains are relevant and provide relevant defences to the infrastructure being assessed.
- *It is held in high regard with the stakeholders and industry* – e.g. the control list should be an accepted national or international standard.
- *It is relatively up-to-date such that it addresses current vulnerabilities* – e.g. the control list addresses the current threats to infrastructure security.

Step 3 – Determine whether all controls are appropriate, and how to deal with conflicts against the control set and the infrastructure. Not all controls will apply to all infrastructures. For example, an external firewall is not necessary if the network is airgapped. If a control is not relevant this will normally mean that

the infrastructure is secure against the attacks that the control is used to defend against. Continuing the example, the airgapped network is not vulnerable to attacks from external networks.

Step 4 – Assess whether each identified control has been implemented, and whether the level of implementation is appropriate and adequate. This is checked against each control in the list, one at a time. For control lists that give multiple sub-controls (as with the CSC 20, discussed in Section 4), it may be desirable to further detail the assessment of the infrastructure's compliance with the control. For example, using logging:

- No Logging: No logging takes place.
- Local Logging: Logging exists but it is basic and localised (e.g. occurring in Windows Event Viewer rather than a dedicated application).
- Centralised Logging: Logging occurs in a standardised output and is centralised (e.g. syslog format is stored in rsyslog).

Step 5 – Combine individual control assessments to give an overall assessment of the infrastructure. Once the status of the relevant controls has been determined, they can be combined to give an overall vulnerability score, a simplified heuristic. If the controls were broken down into sub-controls, the assessment can take this into account using the coverage of the sub-controls to give a finer granularity of vulnerability score. There are certainly more intelligent ways of calculating the network vulnerability score, in a similar way that there are more intelligent ways of assessing the vulnerability of a network than adding up the scores of its vulnerabilities [18]. Determining how to best consider controls to produce an overall and meaningful network vulnerability score is a critical area for future research if controls-based approaches are to be adopted.

4 Applying the Approach to a Scenario using CSC 20

To demonstrate and discuss how our method can be applied in practice, we have applied it to an example network below.

Step 1 – Scope the infrastructure to be evaluated. The example network that we are going to be analysing is a small network for 20 IT professionals. It is primarily a Windows network, with two Windows 2008 Domain Controllers (DC) and 20 Windows 7 laptops which can either connect directly to the network (if they are in the office), or connect to the network via a VPN (if they are outside of the office). As well as these, there are several internal servers offering services to the staff, including file storage and bespoke applications for help with report writing, issue tracking and code repositories. The network is connected to the Internet which is protected by a firewall, and within the office there is a physical Ethernet connection.

Step 2 – Select the control list to be applied. The control set that we will be applying to this network is the CSC 20 [12]. The CSC 20 offers a comprehensive list of low-level controls that are sub-divided into further sub-controls. These controls are up-to-date at the time of the assessment, and are appropriate to the infrastructure that we are assessing.

Step 3 – Determine whether all controls are appropriate, and how to deal with conflicts against the control set and the infrastructure. While most of the controls from the CSC 20 are valid and can be applied to our network, CSC 7: Wireless Access Control is not applicable as the network does not have wireless access. As this control is aimed at preventing unauthorised access from wireless connections it is assessed that the network does not require this control, and can be assumed secure against associated attack vectors.
Step 4 – Assess whether each identified control has been implemented, and whether the level of implementation is appropriate and adequate. While it is not possible to fully assess all CSC 20 controls within this paper, we assess the implementation of one control to demonstrate how the assessment is performed.
Within the CSC 20, each control can be further divided into sub-controls. For example, CSC 1 - Inventory of Authorised and Unauthorised Devices, is divided into 7 sub-controls. These range from having an automated asset discovery tool (CSC 1-1) to using client certificates to validate and authenticate systems prior to their connection to the network (CSC 1-7). Each of these sub-controls has a property relating to their difficulty to implement. Sub-controls are listed as 'quick wins', visibility and attribution, configuration and hygiene, or advanced.
There are many ways these sub-controls can be combined to give the overall effectiveness of the defence against the vulnerabilities they are mitigating. While there are several attempts at accumulating vulnerabilities' risk to give an overall assessment [6][25][27][40], no equivalent has been proposed for controls.
The method we will use for this example is simply to divide the sub-controls in place by the total sub-controls for that control. For example, if there are seven sub-controls for a control, and four of them are in place, that control will receive a score of 0.57 (4/7). If there are nine sub-controls and eight of them are in place, that control will receive a score of 0.89 (8/9). Using this method, an individual score for all of the CSC 20 controls can be quickly calculated.
To assess the 19 remaining CSC security controls we must go through each of them in turn. To give an example of the system, we will cover the controls in CSC 5: Malware Defence.
CSC 5: Malware Defence looks at the different defences that are in place to defend against malware. It has 11 sub-controls. A full listing of CSC 5 can be found online [10].
The example network has thorough anti-malware practices in place. On their workstations, DCs and other servers they have an enterprise malware solution that is automatically updated and has receives new signatures periodically. This software automatically scans email attachments and the contents of removable media before they can be opened. In addition to this, they have both a behavioural and signature-based Intrusion Detection System (IDS). See Table 1 for the completed CSC 5: Malware Defence assessment sheet.

Table 1. CSC 5: Malware Defence

5-1	Yes	The network meets this requirement as they have the necessary antivirus and host-based functionality deployed.
5-2	Yes	Antivirus signatures are pushed out from a centralised repository.
5-3	Yes	The systems have been configured to not auto-run content from removable media.
5-4	Yes	The antivirus software has been configured to automatically scan removable media.
5-5	Yes	Emails going into the organisation are scanned for malicious content before the user receives them.
5-6	Yes	Address Space Layout Randomisation (ASLR) and Data Execution Prevention (DEP) are enabled by default on Windows 7 and Server 2008.
5-7	No	There is no system in place for monitoring the use of external devices, so this requirement is not met.
5-8	Yes	Both signature-based and behavioural IDS are running on the network.
5-9	No	Although there are IDS in place, they only generate alerts when malware is detected, and do not actively prevent its delivery.
5-10	No	There is no incident response process in place for unrecognised malware.
5-11	Yes	DNS query logging is part of their IDS solution.
		CSC 5: Malware Defence Score: 0.727 (Yes - 8 / No - 11)

Table 2. Control Score

CSC 1: 0.8	Inventory of Authorised and Unauthorised Devices
CSC 2: 0.74	Inventory of Authorised and Unauthorised Software
CSC 3: 0.545	Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers
CSC 4: 0.777	Continuous Vulnerability Assessment and Remediation
CSC 5: 0.727	Malware Defences
CSC 6: 0.5	Application Software Security
CSC 7: N/A (1)	Wireless Access Control
CSC 8: 0.666	Data Recovery Capability
CSC 9: 0.75	Security Skills Assessment and Appropriate Training to Fill Gaps
CSC 10: 0.5	Secure Configurations for Network Devices such as Firewalls, Routers, and Switches
CSC 11: 0.5	Limitation and Control of Network Ports, Protocols, and Servers
CSC 12: 0.545	Controlled Use of Administrative Privileges
CSC 13: 0.75	Boundary Defence
CSC 14: 0.925	Maintenance, Monitoring, and Analysis of Audit Logs
CSC 15: 0.667	Controlled Access Based on the Need to Know
CSC 16: 0.235	Account Monitoring and Control
CSC 17: 0.4	Data Protection
CSC 18: 0.333	Incident Response and Management
CSC 19: 0.688	Secure Network Engineering
CSC 20: 1	Penetration Tests and Red Team Exercises
	Final CSC 20 Vulnerability Score: 0.653

Step 5 – Combine individual control assessments to give an overall assessment of the infrastructure. Once the analysis is complete, the scores of each control are combined to give the overall vulnerability score of the example network. The overall vulnerability score for the entire example network is 0.639. See Table 2 for the completed CSC assessment sheet, with values for each CSC control.

This is a simplified example, but demonstrates how control lists can be used to provide a potentially useful metric of vulnerability. This could easily be converted into a Low, Medium and High rating (as is often used for penetration testing and vulnerability scanning) or Fail, Partial and Pass rating (as is often used for high level assessments) depending on how the organisation prefers the information to be presented and how it is going to be used.

5 Reflection on the Approach, its Utility, and Acceptance

The aim of this paper is to explore the use and value in adopting a controls-based approach to assessing infrastructure vulnerability, as opposed to the traditional methods based on detecting and counting low-level vulnerabilities. Below, we reflect on the advantages and limitations of our approach, and compare it against traditional analyses of low-level vulnerability. While this comparison is not exhaustive, it provides a critical reflection on the approach, including its benefits and limitations, and situations where it may be most applicable.

5.1 Potential Advantages of Controls-Based Approach

1) More vulnerabilities exist than controls – While there many known vulnerabilities [14], the majority of these can be mitigated with a competitively small number of controls. Indeed, the ASD strongly argue that 85% of targeted cyber intrusions can be stopped with 4 controls [4]. This is because controls have a one-to-many relationship with vulnerabilities, with one control mitigating or removing many different vulnerabilities. For example, a 'No Execute' bit will prevent many memory injection vulnerabilities.

2) It is quicker and more efficient to determine the presence of controls than vulnerabilities – Finding the vulnerabilities present in a system is a time consuming and subsequently, often time-limited process. This is because of the large number of vulnerabilities that could exist over an infrastructure. Regardless of the infrastructure being tested, there will often be more potential vulnerabilities than it is possible to check for [24].

In penetration testing, the time limitation is expressed in the time scoped for the test. Whoever is scoping the assessment will judge how long the test will need, and then the testers attack the network until that time is up. Penetration testing gives diminishing returns over time. Given a five day test, many vulnerabilities will likely be found over the first one or two days, and fewer will be discovered towards the end of the week. While many vulnerabilities are easy to test for (in part because of vulnerability scanners, which are themselves part of

a penetration tester's tool kit) the more esoteric vulnerabilities can take longer to manually test for, and the majority of any in-depth penetration test will be spent testing for these. With vulnerability scanning, the time is limited by the signatures available to the scanner. When vulnerabilities are discovered, signatures testing that vulnerability are written. These are generally a non-malicious payload sent to the service to observe its response. These signatures take time to write, and there are no vulnerability scanners that claim to find all vulnerabilities.

3) It is less risky to measure the presence of controls than vulnerabilities – While vulnerability scanners can scan for the existence of many vulnerabilities, there are exploits that it is either not possible, or not desirable, to test for (e.g. denial of service vulnerabilities on live systems). This is in contrast to controls, which have all been deliberately implemented by administration staff, ensuring that someone is always aware of their presence. Related to that is the issue that scanning or testing for vulnerabilities can have side effects. Performing scans of devices, even relatively benign scans such as port scanning, can cause device issues and crashes. More advanced tests (e.g. testing vulnerability payloads) can cause increasingly complex problems, such as data corruption or putting a system into an unknown state. Malware, in particular, can result in unwanted side effects, and is a tool not often used in penetration tests (especially against live systems) for this reason. Although this risk can be addressed by taking a virtual image of the infrastructure to test against against rather than the live system, this can be a complex and costly process. As a result, such virtualisation is not performed routinely for vulnerability scanning or penetration testing.

4) Vulnerability-based risk assessment does not consider unknown vulnerabilities – Zero day exploits will not be found by vulnerability scanners, and are unlikely to be detected during penetration tests (depending on the nature of the vulnerability, and the skill, detection and time of the testers). In contrast, controls can and do protect against zero day attacks. For example, Address Space Layout Randomisation will protect against a buffer overflow attack whether the vulnerability it is exploiting is known or not. As a result, the vulnerability of a system to zero day attacks is better measured by the controls it has in place, than its vulnerability to other exploits.

5.2 Potential Limitations of Controls-Based Approach

1) The vulnerability landscape is constantly changing – New attacks (or even whole classes of attacks) can be discovered, and an assessment methodology should be flexible and able to take this into account. Most methodologies used in practice attempt to do this by manually reassigning their assessment criteria (or controls) periodically. With ISO 27001 this happened in 2013, with the update from ISO 27001:2005 to ISO 27001:2013 making changes to the controls (as well as the broader assessment methodology) that were included in the assessment. While the control schemes we have discussed aim to take this into account, there still needs to be a manual update and new version of the control

list in order to do so.
2) Current methods of assessment already have traction and assessment within industry – Penetration tests, vulnerability scans, and high-level information security audits are already established within industry. There already exist trained auditors and organisations who can perform these assessments, and there is a demand and an acceptance of them within business. Compared to low-level vulnerability-based risk assessments, using controls at this level is a relatively unexplored idea. Low-level controls-based risk assessment needs a large amount of development to give it the same maturity as the existing approaches.
While a lack of precedent may be fatal in other areas of organisational decision making, it is an even greater problem within the security industry. Security status is often proven via certification or accreditation, and performing a new approach that does not offer this greatly reduces the benefit of performing the security evaluation in the first place.
3) Vulnerabilities with no known controls are not taken into account, regardless of their presence – New attack vectors with no known controls are not taken into consideration during a controls-based assessment. A good recent example of that is BadUSB [36], which opened up an entirely new line of attacks that had previously not been considered. As a result of this, there are few (if any) controls in place to mitigate against it. This is in contrast with traditional penetration testing and vulnerability scanning, which can consider new vulnerabilities immediately upon their discovery. We saw this with Heartbleed [17], where vulnerability scanners were available to check for its presence the same day that the vulnerability became public knowledge.
Control lists have to be manually updated with a new version produced before a vulnerability is taken into consideration. Generally speaking, this is a bigger weakness against new attack vectors (e.g. BadUSB [36]) than against new vulnerabilities (e.g. Hearthbleed [17]), as new vulnerabilities are often mitigated by existing controls (due to the nature of the one control to many vulnerability relationship). The control lists mentioned in this paper are attempting to stay up-to-date with currently vulnerabilities, but this process is not always easy, and will never be immediate.
4) More research is needed to find suitable methods for determining overall vulnerability – Simply adding the number of scores together to give a value is not as thoroughly researched as many of the referenced vulnerability-based approaches. This is inevitable given the amount of work already in this area. Ideally, to take our work forward, some research and experimentation of that depth should be repeated based on a controls-based approach.

5.3 Approach Acceptance

Ultimately, assessments of infrastructure vulnerability are often driven by business interests. While an organisation may be able to 'sell' a new evaluation methodology, this is not achievable or desirable from an academic perspective. In order for any method of evaluating security to be successful, organisations must see benefit in adopting it. In reality, the vulnerability and risk assessment

market is dominated by established vendors, who have little incentive to change their methodology.
One possible motivator for change is that the standard ways of assessing low-level vulnerability (penetration testing and vulnerability scanning) are expensive. As the majority of the proposed academic approaches [1][18][21] that we have found are based on having pre-existing knowledge of the low-level vulnerabilities in place, this cost exists with them as well. This expense is due to the high-level of assessor skill required to correctly find and identify the vulnerabilities. Even with vulnerability scanners, while they are easy to operate, they often report false positives which require verification to be certain of their existence [42]. This high skill requirement is expensive, and one reason why penetration testing is performed so sparingly.
The expense of finding low-level vulnerabilities contrasts with performing assessments based on control lists. As seen with higher-level assessment (such as ISO 27001) it is possible for assessors to ensure that controls are in place without being highly skilled in the technology they are assessing. It takes a comparatively small amount of training for assessors to be able to correctly identify that controls are in place, and ensure that they are configured correctly. This difference in required knowledge could result in controls-based assessments being more cost effective to perform, and therefore make good business sense, and could help the approach be applied to small and medium sized enterprises (which often have difficulty justifying the resources required to perform security assessments [2]).

6 Conclusion and Future Work

A repeatable and accepted method of judging the low-level vulnerability of an infrastructure would be a useful tool in ensuring system and network security. We argue that one achievable way to do this is transitioning from vulnerability-based risk assessment to controls-based risk assessment. To this end, we proposed an approach based on controls to assess the vulnerability of a computer infrastructure. We then illustrated with an example, how our approach could be applied using a control set, namely the CSC 20, to assess an infrastructure's vulnerability. This presents a simple but effective method of using the sum of all offered (sub-)controls to measure the overall control coverage.
The main difference between our method and earlier approaches is that prior methods attempt to rate overall vulnerability based on the number of low-level vulnerabilities found. We have discussed in depth why we believe this may not be the ideal approach to the situation. While the method that we propose does have drawbacks, we believe these are outweighed by the benefits. Detecting all vulnerabilities is simply not possible, therefore measuring the controls is more likely to give an accurate and achievable indication of true vulnerability.
In terms of future work, there are many avenues to pursue, particularly around demonstrating the validity of this process. As seen with vulnerability-based assessments, there are many different ways that the measured data can be combined to give an overall evaluation of the risk to the network. A first step would

be to apply our approach to real networks, and see how well the results correlate against other risk assessment methodologies. This approach could be taken further by performing a similar assessment to Holm *et al.* [18], comparing different methodologies against the time-to-compromise of a known system. This methodology could also be used to analyse how to best determine the vulnerability scores that our final assessment is based on. At this stage, our research is primarily exploring the advantages to using a controls-based measurement over a vulnerability-based one; repeating their work considering controls-based assessments against vulnerability-based assessments, and comparing those to time-to-compromise would be a key indicator of the validity of this approach.
Similarly, work should be conducted on the optimal way to combine compliance over multiple controls to calculate a realistic vulnerability assessment. An example of this would be looking at whether certain controls should be weighted differently, or how a single major control failure should impact the overall assessment of the infrastructure (i.e. the validity of the weakest link security model in this context). In the example network (Section 4) we merely give a mean of the number of controls that have been complied with. This is clearly a simplistic approach, and is unlikely to give the strongest indicator of the vulnerability of the network. This is similar to other work which has been performed on assessing overall vulnerability by the presence of low-level vulnerabilities [6][25][27][40]. One potentially viable way to do this would be to combine it with control lists that have different levels of controls, e.g. the CSC 20 with 'quick win' controls versus 'advanced' controls. It would be interesting to see which of these give a better indication of the vulnerability of the network, and how this correlates to the cost (both in time and resources) in implementing them.
There should also be further research into different control sets and their efficacy. While the CSC 20 and ASD 35 were discussed in this paper, research should analyse other control sets and how they are generated, to determine what actually makes an effective set of controls. We should study which controls are the most effective at increasing time-to-compromise, and how controls can be combined to offer the most security.
Although this work is mainly intended to be an exploration into the utility of controls-based, low-level risk assessment, we believe there could be genuine benefit to exploring it further. Given the amount of work that has gone into vulnerability-based risk assessment there is still a long way to go to reach that level of maturity.

References

1. Ahmed, M.S., Al-Shaer, E., Khan, L.: A novel quantitative approach for measuring network security. In: INFOCOM 27th Conference on Computer Communications. IEEE (2008)
2. Allan, C., Annear, J., Beck, E., Van Beveren, J.: A framework for the adoption of ICT and security technologies by SMEs. In: 16th Annual Conference of Small Enterprise Association of Australia and New Zealand. vol. 28, pp. 65–81 (2003)

3. Austrailian Signals Directorate - Strategies to Mitigate Targetted Cyber Intrusions [Online] (2014), www.asd.gov.au/infosec/top35mitigatestrategies.htm
4. Austrailian Signals Directorate - Top 4 Strategies to Mitigate Targetted Cyber Intrusions [Online] (2014), www.asd.gov.au/infosec/top-mitigations/top-4-strategies-explained.htm
5. Bhattacharjee, J., Sengupta, A., Mazumdar, C.: A formal methodology for enterprise information security risk assessment. In: International Conference on Risks and Security of Internet and Systems (CRiSIS). pp. 1–9. IEEE (2013)
6. Boyer, W., McQueen, M.: Ideal based cyber security technical metrics for control systems. In: Critical Information Infrastructures Security, pp. 246–260. Springer (2008)
7. Chakrabarti, A., Manimaran, G.: Internet infrastructure security: A taxonomy. Network, IEEE 16(6), 13–21 (2002)
8. Chen, H., Chen, Y., Summerville, D.H.: A survey on the application of FPGAs for network infrastructure security. Communications Surveys & Tutorials, IEEE 13(4), 541–561 (2011)
9. Penetration Testing with Core Impact Pro [Online] (2014), http://www.coresecurity.com/core-impact-pro
10. Council on Cybersecurity [Online] (2014), www.counciloncybersecurity.org
11. Council on Cybersecurity: The ASD 35 and the Council on CyberSecurity Critical Security Controls [Online] (2014), http://www.counciloncybersecurity.org/bcms-media/Files/Download?id=a681a325-e26c-40f4-ad6e-a34200f79084
12. Council on Cybersecurity: The Critical Security Controls for Effective Cyber Defence, version 5.1 [Online] (2015), http://www.counciloncybersecurity.org/bcms-media/Files/Download?id=a52977d7-a0e7-462e-a4c0-a3bd01512144
13. CPNI: Critical Security Controls guidance [Online] (2014), www.cpni.gov.uk/advice/cyber/Critical-controls
14. CVE Details The ultimate security vulnerability datasource [Online] (2014), www.cvedetails.com
15. Feng, N., Wang, H.J., Li, M.: A security risk analysis model for information systems: causal relationships of risk factors and vulnerability propagation analysis. Information Sciences 256, 57–73 (2014)
16. Geers, K.: Live fire exercise: preparing for cyber war. Journal of Homeland Security and Emergency Management 7(1) (2010)
17. The Heartbleed Bug [Online] (2014), http://heartbleed.com
18. Holm, H., Ekstedt, M., Andersson, D.: Empirical analysis of system-level vulnerability metrics through actual attacks. Dependable and Secure Computing, IEEE Transactions on 9(6), 825–837 (2012)
19. COBIT 4.1: Framework for IT Governance and Control [Online] (2014), www.isaca.org/Knowledge-Center/COBIT/Pages/Overview.aspx
20. ISO/IEC 27001 Information security management [Online] (2014), www.iso.org/iso/home/standards/management-standards/iso27001.htm
21. Jajodia, S., Noel, S.: Topological vulnerability analysis. In: Cyber Situational Awareness, pp. 139–154. Springer (2010)
22. Karger, P.A., Schell, R.R.: Multics Security Evaluation Volume II. Vulnerability Analysis. Tech. rep., DTIC Document (1974)
23. Karger, P.A., Schell, R.R.: Multics security evaluation: Vulnerability analysis. In: 18th Annual Computer Security Applications Conference. pp. 127–146. IEEE (2002)

24. Will vulnerabiliy assessments and penetration testing find all the security vulnerabilities in your systems? [Online] (2014), http://www.krypsys.com/news/will-vulnerability-assessments-and-penetration-testing-find-all-the-security-vulnerabilities-in-your-systems
25. Lai, Y.P., Hsia, P.L.: Using the vulnerability information of computer systems to improve the network security. Computer Communications 30(9), 2032–2047 (2007)
26. Liu, S., Kuhn, R., Rossman, H.: Surviving Insecure IT: Effective Patch Management. IT Professional 11(2), 49–51 (2009)
27. McQueen, M.A., Boyer, W.F., Flynn, M.A., Beitel, G.A.: Time-to-compromise model for cyber risk reduction estimation. In: Quality of Protection, pp. 49–64. Springer (2006)
28. NIST: National vulnerability database [online] (2014), http://nvd.nist.gov
29. OpenVAS Open Vulnerability Assessment System [Online] (2014), http://www.openvas.org
30. SANS: 90% of SANS Survey Respondents Are Adopting, or Plan to Adopt, the Critical Security Controls [Online] (2014), http://www.counciloncybersecurity.org/articles/90-of-sans-survey-respondents-are-adopting-or-plan-to-adopt-the-critical-security-controls-2
31. SANS Critical Security Controls for Effective Cyber Defence [Online] (2014), http://www.sans.org/critical-security-controls
32. Schneier, B.: Schneier on Security: The Internet of Things Is Wildly Insecure And Often Unpatchable [Online] (2014), http://www.schneier.com/essays/archives/2014/01/the_internet_of_thin.html
33. Shah, S., Mehtre, B.: An overview of vulnerability assessment and penetration testing techniques. Journal of Computer Virology and Hacking Techniques pp. 1–23 (2014)
34. Snort [Online] (2014), http://www.snort.org
35. Sommestad, T., Ekstedt, M., Holm, H.: The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures. Systems Journal, IEEE 7(3), 363–373 (2013)
36. Bad USB [Online] (2014), srlabs.de/badusb/
37. Szwed, P., Skrzyński, P.: A new lightweight method for security risk assessment based on fuzzy cognitive maps. International Journal of Applied Mathematics and Computer Science 24(1), 213–225 (2014)
38. Tenable Network Security Nessus [Online] (2014), http://www.tenable.com
39. Thompson, K.: Reflections on trusting trust. Communications of the ACM 27(8), 761–763 (1984)
40. Tupper, M., Zincir-Heywood, A.N.: VEA-bility security metric: A network security analysis tool. In: Third International Conference on Availability, Reliability and Security (ARES). pp. 950–957. IEEE (2008)
41. Valli, C., Woodward, A., Hannay, P., Johnstone, M.: Why penetration testing is a limited use choice for sound cyber security practice. In: Proceedings of the Conference on Digital Forensics, Security and Law. pp. 35–40 (2014)
42. Vieira, M., Antunes, N., Madeira, H.: Using web security scanners to detect vulnerabilities in web services. In: International Conference on Dependable Systems & Networks (DSN). IEEE/IFIP. pp. 566–571. IEEE (2009)