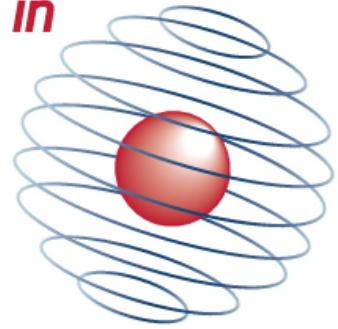




CENTRE *for* DOCTORAL TRAINING *in*
**CYBER
SECURITY**



CDT Technical Paper

02/14

**Controls-Based Assessment of
Infrastructure Vulnerability**

Oliver Farnan

Controls-Based Assessment of Infrastructure Vulnerability

A structured method for assessing technical infrastructure vulnerability based on control lists instead of vulnerability lists

Oliver J. Farnan
CDT for Cyber Security
Oxford University
Oxford, UK
oliver.farnan@balliol.ox.ac.uk

Jason R. C. Nurse
Cyber Security Centre
Oxford University
Oxford, UK
jason.nurse@cs.ox.ac.uk

Abstract—Assessing the vulnerability of an enterprise’s infrastructure is an important step in judging the security of a network and the trustworthiness and quality of the information that flows through it. Currently real-world infrastructure vulnerability is often judged in an ad hoc manner, based on the criteria and experience of the assessors. While methodological approaches to assessing infrastructure vulnerability exist, in practice they are not academically rigorous, having grown organically to meet business requirements. Our aim in this paper therefore is to study infrastructure vulnerability from a more structured perspective. We introduce and explore a novel way of assessing computer network infrastructure 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 look at how one of them, namely the Council on Cyber Security’s Top 20 Critical Security Controls, can be applied.

Keywords—*infrastructure vulnerability; controls; technical audit; network security; risk assessment; penetration test; vulnerability scan; information security;*

I. INTRODUCTION

Finding vulnerabilities is a difficult and arduous task ^[31]. Despite this, assessing the vulnerability of computer networks to attacks is a common and necessary enterprise security activity. This includes two major types of vulnerability assessment: low-level (e.g. vulnerability scans looking for vulnerabilities such as MS08-067¹) and high-level (e.g. ISO 27001 ^[15] audits looking for the existence of appropriate cryptographic storage). Both are required for various types of security risk assessments, which assist in critical business decisions, help demonstrate an organisation’s security credentials, and determine the risk of compromise to information contained within or sent through a network. 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 network. A common set of standards (and one which we will use as an example in this paper) is ISO 27001 ^[15].

This paper aims to explore the implementation of a structured and rigorous approach to assessing the vulnerability of a computer network’s infrastructure against low-level vulnerabilities (e.g. whether its OpenSSL implementation is vulnerable to Heartbleed ^[30]). Our methodology 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 techniques to present on (e.g. ISO 27001) and instead intentionally focus on the low-level technical vulnerability of an infrastructure. Moreover, we ignore higher-level factors (e.g. outside threats, processes users and administrators follow when interacting with the network, etc.) and do not attempt to judge the overall risk to the infrastructure.

Basing our methodology on 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][14][16]}. Similarly, within businesses, assessment of low-level vulnerability is performed by vulnerability scanning or penetration testing to discover existing vulnerabilities on the network ^{[17][18]}.

This low-level evaluation contrasts with higher-level risk assessment methodologies, whose assessments are based on the existence (or lack thereof) of procedural controls that an organisation has in place. Such higher-level methodologies are more structured and well established. 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

¹ MS08-067 is a low-level vulnerability in the Windows Server Service that allows full remote code execution when sent a specially crafted RPC request. It affects Windows 2000, XP, Server 2003, Vista, and Server 2008.

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 the methodology we are exploring, 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 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 ^{[6][7][20]}.

Despite the need to assess low-level network vulnerability, there is not a generally accepted way of performing it. Different methods exist for measuring the higher-level security of a network or system (such as ISO 27001), but there is no established way of providing a simple low-level and unbiased assessment of the infrastructure's vulnerability. On one hand there exists technical testing such as vulnerability scanning tools (e.g. Nessus ^[28]) and penetration testing techniques, which are widely employed but often unstructured and incomplete. 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 give an example (Section 4). We then reflect on our proposal (Section 5), draw conclusions (Section 6) and discuss areas for future research (Section 7).

II. RELATED WORK

Currently, there are several ways organisations attempt to assess their infrastructure's vulnerability. Typically these involve some kind of audit and risk assessment being carried out. These can range from comparing the infrastructure to standard procedural checklists (such as the ISO 27000 family), to hands on technical audits (such as 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.

A real problem assessors face when evaluating network security is that the assessment will 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 assessed affect the assessment of that which they are currently auditing. To combat this, at the high-level, strict information security assessment methodologies have been created such as ISO 27001 ^[15] and COBIT ^[8]. These aim to provide a rigid framework for assessors to follow, 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.

While these assessment methodologies are in place for the procedural side of information security, they are less rigorous 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 assessment reports of this kind focus on specific low-level vulnerabilities (e.g. remote-code execution vulnerability MS08-067 found on the domain controller) and do not have a structured way of building on these to provide an overall assessment of the vulnerability of the infrastructure. Executive summaries are often included to provide this bigger-picture assessment, but as there is no well-defined process it is inevitably tainted by the testers' previous experience.

Penetration testing has been used for a long time, and began with Karger and Schell's evaluation of Multics for the US Air Force ^[17]. They performed a thorough evaluation of Multics security, and found several ways to bypass its multi-level user control system. Penetration testing is introduced as 'tiger teaming' within the paper, but 'red teaming' is used in other literature².

Ken Thompson's 'Reflections on Trusting Trust' ^[31] is the classic paper on the difficulty on detecting vulnerabilities. He demonstrates the possibility of a vulnerability invisible to even thorough source code review, and gives a clever example of a vulnerability that is 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 vulnerability scanners include Nessus ^[28], OpenVAS ^[23] and Core Impact ^[24], and they are used in many proposed academic methods for attempting to assess infrastructure vulnerability ^{[14][16]}.

In practice, both vulnerability scanners and penetration tests 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 (or at least payloads that do not leave lasting damage). Vulnerability scanning involves scanning 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 that attack it is recorded and presented in a report.

The vulnerabilities themselves are recorded in vulnerability databases such as CVE Details ^[11] and the NIST National Vulnerability Database ^[22]. These databases store details of a vast collection of vulnerabilities for various systems, along with the details of the vulnerabilities such as at risk software versions and the effect of the compromise. The standard method of measuring the impact of these vulnerabilities is by using a CVSS score ^[33], which gives a metric based on factors such as ease of exploitation and the effect on the

² Within the penetration testing community 'red teaming' refers to a specific type of penetration test, normally involving a broadly defined scope of targets and attack surface.

confidentiality, integrity and availability of the vulnerable system. Vulnerability scanners list detected vulnerabilities by their reference in vulnerability databases, and include a risk rating either based on or similar to a CVSS score.

There has also been a noteworthy amount of academic work on assessing infrastructure vulnerability. These primarily aim to judge the vulnerability of a network based on the low-level vulnerabilities detected. Jajodia and Noel^[16] 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 based on vulnerability scanners such as Nessus^[28] and can be used with the Intrusion Detection System Snort^[27] to correlate received alerts. Ahmed, Al-Shaer and Khan^[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, Ekstedt and Andersson^[14] look at the effectiveness of different methods of judging system vulnerability through CVSS scores. They base their work on the time taken to compromise systems in the cyber-defence exercise Baltic Shield^[13], 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.

To summarise the research above, all of these proposed methods are based on the vulnerabilities present. Despite these proposals, none of these methodologies have gone into widespread use on live systems. This is in contrast to the higher-level vulnerability assessment methodologies, which are primarily control-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 test environments, they are likely to be more difficult to use to analyse 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 of this are the Council on Cyber Security’s Top 20 Critical Security Controls^[9] and Australian Signals Directorate (ASD) 35 Cyber Security Mitigation Strategies^[2]. Both of these contain lists of low-level controls that can be applied to a computer network.

The Council on Cyber Security’s Top 20 Critical Security Controls (20 CSC) are a list of 20 security controls created based on an observed need for a standardised controls list of this type. They were drawn up by a broad collection of ‘companies, government agencies, institutions, and individuals from every part of the [security] ecosystem’^[9]. While security

policy is often written by those working in defensive security, these controls were created with the help of offensive security actors, using an ‘offense informs defence’ approach. They have been widely adopted by other organisations^[35], including SANS^[25] and the UK Centre for the Protection of National Infrastructure (CPNI)^[10]. The CSC 20 are regularly updated, and are on version 5 since their 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 are a series of 35 controls ranked in order of overall effectiveness at network protection. They place particular emphasis on their top 4 mitigation strategies, which they argue stop 85% of targeted cyber intrusions^[2]. They are written at a slightly lower-level of abstraction than the CSC 20, and largely include similar overlapping controls^[29]. 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*).

We believe these and similar control lists offer a better opportunity to assess overall network vulnerability than measuring the vulnerabilities themselves. They recommend controls that match up well against the low-level vulnerabilities that infrastructures commonly face, and as such, we believe offer a well-defined and achievable way to properly assess infrastructure vulnerability.

III. APPROACH – GENERAL

Typically, the assessment of infrastructure vulnerability is based upon the correct identification of the vulnerabilities present. This can be performed either with low-level vulnerabilities, as identified with vulnerability scanning or penetration testing (e.g. testing for MS08-067), or with higher level vulnerabilities (e.g. data-in-transit is unencrypted and vulnerable to interception). This is the method used for both vulnerability scanning and penetration testing.

Our methodology does not attempt to measure vulnerability in this way. Instead, we assume all infrastructures to be vulnerable, and that this vulnerability can only be mitigated with the implementation of certain controls. This is similar to higher-level risk assessment methodologies (such as 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^{[6][7]}. Current devices are often not built with security in mind, and contain numerous undiscovered vulnerabilities^[26]. 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 methodology relies on lists of controls

compiled from other sources. We see two primary advantages for doing this. Firstly, the methodology itself can be static and is not constantly changing. This makes the process more robust and not dependent on specific vulnerabilities, and not impacted by the changing vulnerability landscape. Secondly, there is plenty of research into the most effective controls to address vulnerability. Instead of replicating this work we can leverage 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 multistep methodology. We envision 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.* It is important to determine exactly which infrastructure assets are to be included into the assessment.

Step 2 – *Select the control list to be applied.* Compliance with different control lists will offer different levels of assurance to the infrastructure. Control schemes should be chosen with the following criteria:

- Meets the requirements of the assessment;
- Is appropriate to the infrastructure;
- Held in high regard with the stakeholders and industry;
- Is relatively up-to-date such that it addresses current vulnerabilities.

Step 3 – *Determine whether the specific controls are appropriate.* 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 or adequate.* This is done against each control in the list, systematically. For control lists that give multiple sub-controls (as with the CSC 20, discussed in Section 4) it may be desirable to further break down 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 added up to give an overall vulnerability score here, a simplified heuristic. If the controls were broken down into sub-controls, this assessment can take this into account using the coverage of the sub-controls to give a finer granularity of vulnerability score.

There are almost certainly more intelligent ways of calculating the vulnerability score of the network, 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^[14]. Working out how to best consider controls to give an overall vulnerability work is a critical area for future research if control-based approaches are to be advocated over vulnerability-based approaches.

IV. APPLYING THE APPROACH TO AN EXAMPLE SCENARIO USING CSC 20

To demonstrate how our method can be applied in practice, we have applied the Council on Cyber Security's CSC 20 to an example network below.

Step 1 – *Scope the infrastructure to be evaluated.*

The example network we are going to be applying the CSC 20 to 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 set to be applied.*

The control set that is being applied is the CSC 20. 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 this attack vector.

Step 4 – *Assess the implementation of each control.*

While it is not possible to fully assess all CSC 20 controls within this paper, we assess the implementation of two controls to demonstrate how the assessment is performed.

Within the CSC 20, each control can be further subdivided into sub-controls. For example, *CSC 1 – Inventory of Authorised and Unauthorised Devices*, is subdivided 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 increase 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 ^{[5][19][21][32]}, 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 security controls we must go through each of them in turn. To give an example of the system, we will cover the controls *CSC 5: Malware Defence* and *CSC 16: Account Monitoring and Control*.

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 in the appendix.

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 new signatures pushed out periodically. This software automatically scans email attachments and the contents of removable media before they can be opened. As well as this they have both a behavioral and signature-based IDS. We therefore have the following assessment sheet.

use of external devices, so this requirement is not met.	
CSC 5-8: Both signature-based and behavioral IDS are running on the network.	Yes
CSC 5-9: Although there are IDS in place, they only generate alerts when malware is detected, and do not actively prevent its delivery.	No
CSC 5-10: There is no incident response process in place for unrecognised malware.	No
CSC 5-11: DNS query logging is part of their IDS solution.	Yes
Total	0.727

Altogether there were a total of eight passes and three fails under *CSC 5: Malware Defence*. This gives a total vulnerability score of 0.727.

The second example control is *CSC 16: Account Monitoring and Control*, which looks at how the organisation manages user accounts. It has 17 sub-controls. A full listing of CSC 16 can be found in the appendix.

The example network has weaker account management practices in place. Although they use good basic Windows domain security, the user account management on their additional servers and bespoke services is not as strong, meaning that they fail many of these controls.

CSC 16: Account Monitoring and Control	Deployed
CSC 16-1: All system accounts are tied to an owner.	Yes
CSC 16-2: Although Windows domain accounts have an expiration date, this is not the case for the accounts on other servers and services. As a result this control is failed.	No
CSC 16-3: Although Windows domain accounts are set to automatically create a report, this is not the case for accounts on other servers and services. As a result this control is failed.	No
CSC 16-4: Although a process is in place to remove Windows domain accounts when employees leave, this does not take accounts belonging to other servers and services into consideration. As a result this control is failed.	No
CSC 16-5: Although Windows domain accounts are regularly monitored, this is not the case for accounts on other servers and services. As a result this control is failed.	No
CSC 16-6: All systems are configured to lock the screen after a set period of time.	Yes
CSC 16-7: Although Windows domain accounts are monitored for dormant accounts, this is not the case for accounts on other servers and services. As a result this control is failed.	No
CSC 16-8: Although Windows domain accounts require strong passwords and periodic changes, this is not the case for accounts on other servers and services. As a result this control is failed.	No
CSC 16-9: Although Windows domain accounts automatically lock out after several failed login attempts, this is not the case for accounts on other servers and services. As a result this control is failed.	No
CSC 16-10: Although Windows domain accounts match employees to accounts, this is not the case for accounts on other servers and services. As a result this control is failed.	No
CSC 16-11: Although Windows domain accounts are logged for attempted logins on deactivated accounts,	No

CSC 5: Malware Defence	Deployed
CSC 5-1: The network meets this requirement as they have the necessary antivirus and host based functionality deployed.	Yes
CSC 5-2: Antivirus signatures are pushed out from a centralised repository.	Yes
CSC 5-3: The systems have been configured to not auto-run content from removable media.	Yes
CSC 5-4: The antivirus software has been configured to automatically scan removable media.	Yes
CSC 5-5: Emails going into the organisation are scanned for malicious content before the user receives them.	Yes
CSC 5-6: ASLR and DEP are enabled by default on Windows 7 and Server 2008.	Yes
CSC 5-7: There is no system in place for monitoring the	No

this is not the case for accounts on other servers and services. As a result this control is failed.	
CSC 16-12: Although Windows domain accounts are performed through Active Directory, this is not the case for accounts on other servers and services. As a result this control is failed.	No
CSC 16-13: There is no profiling of user accounts to determine unusual usage patterns.	No
CSC 16-14: Multi-factor authentication is used for the VPN connection, but it is not used for any other accounts.	No
CSC 16-15: All the services on the internal network are accessed via the VPN, which sends the credentials over an encrypted channel and stores the password hash securely.	Yes
CSC 16-16: Passwords are not always encrypted before transmission over the network.	No
CSC 16-17: All passwords on all servers and services are correctly stored as hashes, with access restricted to high-privilege administrator accounts.	Yes
Total	0.235

Altogether there were a total of four passes and thirteen fails under *CSC 16: Account Monitoring and Control*. This gives a total vulnerability score of 0.235.

Although many of the controls are in place for some of the services on this network (in particular the Windows domain services) they are not in place for additional services sitting on other servers. This has resulted in a large amount of failures against the sub-controls for this control.

Step 5 – *Combine individual control assessments to give an overall assessment of the infrastructure.*

The scores of each control are now added together and averaged to give the overall vulnerability score of the example network. The overall vulnerability score for the entire example network is 0.639. In the report sheet below, we present the completed study.

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

CSC 17: Data Protection	0.4
CSC 18: Incident Response and Management	0.333
CSC 19: Secure Network Engineering	0.688
CSC 20: Penetration Tests and Red Team Exercises	1
Total	0.653

V. REFLECTION

Our proposal is an exploratory idea, testing the waters regarding the use of controls to assess vulnerability instead of attempting to enumerate the low-level vulnerabilities themselves. While we believe this gives a stronger indication of overall vulnerability, we realise it is a new and unaccepted approach when operating at this level of abstraction.

A. Comparison to Other Academic Approaches

The main difference between our method and other academic approaches is that the proposed academic 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.

One weakness of the approach we have performed is in the method through which the controls should be combined to give an overall security assessment. 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 of that depth should be repeated based on our proposed controls-based approach.

B. Critically Examining the Controls-Based Method

Compared to low-level vulnerability-based risk assessments, using controls at this level is a relatively unexplored idea. Low-level control-based risk assessment needs a large amount of work to give it the same maturity as low-level vulnerability-based risk assessment.

The primary criticism of our method is that vulnerabilities that have no known controls are not taken into consideration, even though they clearly present an increase in risk. Vulnerabilities that were previously unknown, or have no known mitigations, will not be represented in the system's vulnerability rating. A good recent example of that is BadUSB^[4], which opened up an entire 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. Despite this, we believe there is still merit in exploring the possibility of measuring low-level infrastructure vulnerability by the presence (or lack thereof) controls. There are a number of reasons why we believe this approach is suitable:

1) *More vulnerabilities exist than controls.* While there are tens of thousands of known vulnerabilities^[11], the majority of these can be mitigated with a relatively small number of controls. Indeed the ASD believe that 85% of targeted cyber intrusions can be stopped with 4 controls^[3]. 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 into data attacks (presuming no alternate way of circumventing it can be found).

2) *It is quicker to determine the presence of controls than vulnerabilities.* Finding the vulnerabilities present in a system is a time consuming and subsequently time-limited process. This is because there are tens of thousands of vulnerabilities that could be checked for. Regardless of the infrastructure being tested, there will often be more potential vulnerabilities than it is possible to check for ^[34].

With penetration testing, this is made manifest by the time that is scoped for the test. Whoever is scoping the assessment will judge how long the test will need and then the testers will keep on attacking that network until they run out of time. 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 then less and less will be discovered as time goes on. While many vulnerabilities are easy to test for (in part because of vulnerability scanners, which are themselves an essential part of a penetration testers tools) the more esoteric vulnerabilities can take longer to manually test for.

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 fired at the service to test its response. These signatures take time to write, and there are no vulnerability scanners that claim to find all vulnerabilities. Instead (like with controls) signatures are written for the highest priority vulnerabilities as they are found (e.g. recently MS13-002).

Due to the time it takes to measure low-level vulnerabilities, and the many-to-one relationship with vulnerabilities-to-controls, we believe it should be a much quicker process to measure the controls instead of the vulnerabilities. In real world terms, this difference in assessment length will manifest itself in terms of cost to perform the assessment. As a result, controls-based risk assessment should be a cheaper alternative to vulnerability-based risk assessment.

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 many exploits that it is either not possible, or not desirable, to test for (e.g. denial of service vulnerabilities on live systems). Ken Thompson's famous Reflections on Trusting Trust ^[31] gives the classic example of the difficulty in finding vulnerabilities, and BadUSB ^[4] presents a recent one. This is in contrast to controls, which must have all been deliberately implemented by administration staff.

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 frequently cause device issues and crashes. More complex tests such as testing vulnerability payloads can cause more complex issues, such as data corruption and putting a system into an unknown state.

4) *Controls are designed and implemented to mitigate vulnerabilities as they are discovered.* When new vulnerabilities are found, controls are developed to mitigate

them. Some of these controls will be actions that can already be put in place (e.g. forcing TLS 1.1 or 1.2 to protect against BEAST ^[12]), while for some vulnerabilities entirely new controls will have to be developed to counter that vulnerability. The more serious the vulnerability, the more resources are going to be spent by the industry addressing it.

One difficulty with introducing a controls agnostic method of vulnerability assessment is that specific vulnerabilities are constantly changing. New attacks, or even whole classes of attacks, can be discovered. A previously covered example of this is BadUSB, which introduced an entirely unexpected attack vector ^[4]. Any assessment methodology must be flexible and able to take this into account. Most methodologies used in practice attempt to do this by manually reassigning their assessment criteria periodically. With ISO 27001 we witnessed this in 2013, with the update from ISO 27001:2005 to ISO 27001:2013 making changes to the controls that were included in the assessment.

The control schemes we have introduced take this into account. As new vulnerabilities are discovered, the most effective mitigations to these vulnerabilities are included in the control lists. If vulnerabilities present high risks then controls against that vulnerability are more likely to be included in future revisions of the control lists; if vulnerabilities present low risks then controls against them are going to be included with a lower priority. Similarly, if controls are more effective at stopping the newly discovered vulnerabilities, they are more likely to be included; if they are less effective as controls then they are less likely to be included. As a result, control schemes are based on the controls which have the most effect when addressing vulnerability. One advantage of relying on outside control lists is that this ever changing vulnerability landscape is taken into consideration by the control list, so that the assessment scheme itself does not have to.

5) *Vulnerability-based risk assessment also 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. Having Address Space Layout Randomisation will stop a basic 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 actually better measured by the controls it has in place, than its vulnerability to other exploits.

6) *Controls-based risk assessment is already accepted within information security, with ISO 27001 and other higher-level security assessments.* It is used for these schemes precisely for the above reasons. It is easy to measure what controls have been implemented, and gives a good overview of the security of the system.

C. Approach Acceptance

Ultimately, assessments of this kind are often driven by business interests. While an organisation may be able to 'sell' a new evaluation methodology to clients, 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 the established actors, 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 most of the proposed academic approaches are based on having pre-existing knowledge of the low-level vulnerabilities in place, this cost exists with them too. 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^[33]. This high skill requirement is expensive, and the reason why penetration testing is performed so scarcely.

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 experts in the technology they are assessing. It takes a relatively 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 control-based assessments being cheaper to perform, and therefore make perfect business sense.

VI. CONCLUSION

A repeatable and accepted method of judging the low-level vulnerability of information 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 control-based risk assessment.

We have described one example of how a control set, namely the CSC 20, could be used to assess infrastructure vulnerability. Our proposal is a simple method of using the sum of all offered (sub-)controls to measure the overall control coverage. We recommend that further research is done into the best way to use control lists to give an accurate assessment of vulnerability.

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.

VII. FUTURE WORK

While we have introduced a general approach for measuring infrastructure vulnerability, work must still be done to demonstrate 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 low-level 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, Ekstedt and Andersson [14], comparing different methodologies against the time-to-compromise of a known system. This methodology could also be used to look at how to best determine the vulnerability scores that our final assessment is based on. At this stage our research is primarily pointing out the advantages to using a controls-based measurement over a vulnerability-based one, but repeating their work including controls-based assessments against vulnerability-based assessments, and comparing those to time-to-compromise would be a key indicator of the validity of our suggestion.

Similarly, work should be done 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 one 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 5) 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^{[5][19][21][32]}. One interesting 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, it is worth looking into other control sets and how control sets are generated, to determine what actually makes an effective set of controls. What controls are the most effective at increasing time-to-compromise, and how should controls be combined to produce the best effect.

Although this work is just an exploration into the possibility of control 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 maturity.

REFERENCES

- [1] Ahmed, Mohammad Salim, Ehab Al-Shaer, and Latifur Khan. "A novel quantitative approach for measuring network security." INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. IEEE, 2008.
- [2] Australian Signals Directorate Strategies to Mitigate Targeted Cyber Intrusions [Online]. Available : www.asd.gov.au/infosec/top35mitigationstrategies.htm (Accessed 09/09/2014)
- [3] Australian Signals Directorate – Top 4 Strategies to Mitigate Targetted Cyber Intrusions: Nabdatirt Requirement Explained [Online]. Available: www.asd.gov.au/infosec/top-mitigations/top-4-strategies-explained.htm (Accessed 09/09/2014)

[4] BadUSB [Online]. Available: srlabs.de/badusb/ (Accessed 09/09/2014)

[5] Boyer, Wayne, and Miles McQueen. "Ideal based cyber security technical metrics for control systems." *Critical Information Infrastructures Security*. Springer Berlin Heidelberg, 2008. 246-260.

[6] Chakrabarti, Anirban, and G. Manimaran. "Internet infrastructure security: A taxonomy." *Network*, IEEE 16.6 (2002): 13-21.

[7] Chen, Hao, Yu Chen, and Douglas H. Summerville. "A survey on the application of FPGAs for network infrastructure security." *Communications Surveys & Tutorials*, IEEE 13.4 (2011): 541-561.

[8] COBIT 4.1: Framework for IT Governance and Control [Online]. Available: www.isaca.org/Knowledge-Center/COBIT/Pages/Overview.aspx (Accessed 09/09/2014)

[9] Council on Cyber Security [Online]. Available: www.counciloncybersecurity.org (Accessed 09/09/2014)

[10] CPNI Critical Security Controls guidance [Online]. Available: www.cpni.gov.uk/advice/cyber/Critical-controls (Accessed 09/09/2014)

[11] CVE Details – The ultimate security vulnerability datasources [Online]. Available: www.cvedetails.com (Accessed 09/09/2014)

[12] Duong, Thai, and Juliano Rizzo. "Here come the ninjas." Unpublished manuscript (2011).

[13] Geers, Kenneth. "Live fire exercise: preparing for cyber war." *Journal of Homeland Security and Emergency Management* 7.1 (2010).

[14] Holm, Hannes, Mathias Ekstedt, and Dennis Andersson. "Empirical analysis of system-level vulnerability metrics through actual attacks." *Dependable and Secure Computing*, IEEE Transactions on 9.6 (2012): 825-837.

[15] ISO/IEC 27001 – Information security management [Online]. Available: www.iso.org/iso/home/standards/management-standards/iso27001.htm (Accessed 09/09/2014)

[16] Jajodia, Sushil, and Steven Noel. "Topological vulnerability analysis: A powerful new approach for network attack prevention, detection, and response." *Algorithms, Architectures, and Information Systems Security* (2007).

[17] Karger, Paul A., and Roger R. Schell. "Multics security evaluation: Vulnerability analysis." *Computer Security Applications Conference*, 2002. Proceedings. 18th Annual. IEEE, 2002.

[18] Karger, Paul A., and Roger R. Schell. *Multics Security Evaluation Volume II. Vulnerability Analysis*. No. ESD-TR-74-193-Vol-2. ELECTRONIC SYSTEMS DIV LG HANSCOM FIELD MASS, 1974.

[19] Lai, Yeu-Pong, and Po-Lun Hsia. "Using the vulnerability information of computer systems to improve the network security." *Computer Communications* 30.9 (2007): 2032-2047.

[20] Liu, Kuhn and Rossman. *Insecure, I. T. "Surviving insecure it: Effective patch management."* (2009).

[21] McQueen, Miles A., et al. "Time-to-compromise model for cyber risk reduction estimation." *Quality of Protection*. Springer US, 2006. 49-64.

[22] National Vulnerability Database [Online]. Available: nvd.nist.gov (Accessed 09/09/2014)

[23] OpenVAS – Open Vulnerability Assessment System [Online]. Available: www.openvas.org (Accessed 09/09/2014)

[24] Penetration Testing with Core Impact Pro [Online]. Available: www.coresecurity.com/core-impact-pro (Accessed 09/09/2014)

[25] SANS Critical Security Controls for Effective Cyber Defence [Online]. Available: www.sans.org/critical-security-controls (Accessed 09/09/2014)

[26] Schneier. *Schneier on Security: The Internet of Things Is Wildly Insecure – And Often Unpatchable* [Online]. Available: www.schneier.com/essays/archives/2014/01/the_internet_of_thin.html (Accessed 09/09/2014)

[27] Snort [Online]. Available: www.snort.org (Accessed 09/09/2014)

[28] Tenable Network Security – Nessus [Online]. Available: www.tenable.com (Accessed 09/09/2014)

[29] The ASD 35 and the Council on CyberSecurity Critical Security Controls [Online]. Available: www.counciloncybersecurity.org/bcms-media/Files/Download?id=a681a325-e26c-40f4-ad6e-a34200f79084 (Accessed 09/09/2014)

[30] The Heartbleed Bug [Online]. Available: heartbleed.com (Accessed 09/09/2014)

[31] Thompson, Ken. "Reflections on trusting trust." *Communications of the ACM* 27.8 (1984): 761-763.

[32] Tupper, Melanie, and A. Nur Zincir-Heywood. "VEA-bility security metric: A network security analysis tool." *Availability, Reliability and Security*, 2008. ARES 08. Third International Conference on. IEEE, 2008.

[33] Vieira, Marco, Nuno Antunes, and Henrique Madeira. "Using web security scanners to detect vulnerabilities in web services." *Dependable Systems & Networks*, 2009. DSN'09. IEEE/IFIP International Conference on. IEEE, 2009.

[34] Will vulnerability assessments and penetration testing find all the security vulnerabilities in your systems? [Online]. Available: www.krypsys.com/news/will-vulnerability-assessments-and-penetration-testing-find-all-the-security-vulnerabilities-in-your-systems (Accessed 11/09/2014)

[35] 90% of SANS Survey Respondents Are Adopting, or Plan to Adopt, the Critical Security Controls [Online]. Available: www.counciloncybersecurity.org/articles/90-of-sans-survey-respondents-are-adopting-or-plan-to-adopt-the-critical-security-controls-2 (Accessed 11/09/2014)

APPENDIX

A. CSC 5: Malware Defence

ID #	Description	Category
CSC 5-1	Employ automated tools to continuously monitor workstations, servers, and mobile devices with anti-virus, anti-spyware, personal firewalls, and host-based IPS functionality. All malware detection events should be sent to enterprise anti-malware administration tools and event log servers.	Quick win
CSC 5-2	Employ anti-malware software that offers a remote, cloud-based centralized infrastructure that compiles information on file reputations or have administrators manually push updates to all machines. After applying an update, automated systems should verify that each system has received its signature update.	Quick win
CSC 5-3	Configure laptops, workstations, and servers so that they will not auto-run content from removable media, like USB tokens (i.e., "thumb drives"), USB hard drives, CDs/DVDs, FireWire devices, external serial advanced technology attachment devices, and mounted network shares,.	Quick win
CSC 5-4	Configure systems so that they automatically conduct an anti-malware scan of removable media when inserted.	Quick win
CSC 5-5	Scan and block all e-mail attachments entering the organization's e-mail gateway if they contain malicious code or file types that are unnecessary for the organization's business. This scanning should be done before the e-mail is placed in the user's inbox. This includes e-mail content filtering and web content filtering.	Quick win
CSC 5-6	Enable anti-exploitation features such as Data Execution Prevention (DEP), Address Space Layout Randomization (ASLR), virtualization/containerization, etc. For increased protection, deploy capabilities such as Enhanced Mitigation Experience Toolkit (EMET) that can be configured to apply these protections to a broader set of applications and executables.	Quick win
CSC 5-7	Limit use of external devices to those that have a business need. Monitor for use and attempted use of external devices.	Quick win
CSC 5-8	Ensure that automated monitoring tools use behavior-based anomaly detection to complement traditional signature-based detection.	Visibility/Attribution
CSC	Use network-based anti-malware tools to identify	Visibility/

5-9	executables in all network traffic and use techniques other than signature-based detection to identify and filter out malicious content before it arrives at the endpoint.	Attribution
CSC 5-10	Implement an incident response process that allows the IT support organization to supply the security team with samples of malware running on corporate systems that do not appear to be recognized by the enterprise's anti-malware software. Samples should be provided to the security vendor for "out-of-band" signature creation and later deployed to the enterprise by system administrators.	Advanced
CSC 5-11	Enable domain name system (DNS) query logging to detect hostname lookup for known malicious C2 domains.	Advanced

B. CSC 16: Account Monitoring and Control

ID #	Description	Category
CSC 16-1	Review all system accounts and disable any account that cannot be associated with a business process and owner.	Quick win
CSC 16-2	Ensure that all accounts have an expiration date associated with the account.	Quick win
CSC 16-3	Ensure that systems automatically create a report that includes a list of locked-out accounts, disabled accounts, accounts with passwords that exceed the maximum password age, and accounts with passwords that never expire. This list should be sent to the associated system administrator in a secure fashion.	Quick win
CSC 16-4	Establish and follow a process for revoking system access by disabling accounts immediately upon termination of an employee or contractor. Disabling instead of deleting accounts allows preservation of audit trails.	Quick win
CSC 16-5	Regularly monitor the use of all accounts, automatically logging off users after a standard period of inactivity.	Quick win
CSC 16-6 (NEW)	Configure screen locks on systems to limit access to unattended workstations.	Quick win
CSC 16-7	Monitor account usage to determine dormant accounts, notifying the user or user's manager. Disable such accounts if not needed, or document and monitor exceptions (e.g., vendor maintenance accounts needed for system recovery or continuity operations).	Quick win
CSC 16-8	Require that all non-administrator accounts have strong passwords that contain letters, numbers, and special characters, be changed at least every 90 days, have a minimal age of one day, and not be allowed to use the previous 15 passwords as a new password. These values can be adjusted based on the specific business needs of the organization.	Quick win
CSC 16-9	Use and configure account lockouts such that after a set number of failed login attempts the account is locked for a standard period of time.	Quick win
CSC 16-10	Require that managers match active employees and contractors with each account belonging to their managed staff. Security or system administrators should then disable accounts that are not assigned to active employees or contractors.	Visibility/Attribution
CSC 16-11	Monitor attempts to access deactivated accounts through audit logging.	Visibility/Attribution
CSC 16-12 (NEW)	Configure access for all accounts through a centralized point of authentication, for example Active Directory or LDAP. Configure network and security	Configuration/Hygiene

	devices for centralized authentication as well.	
CSC 16-13	Profile each user's typical account usage by determining normal time-of-day access and access duration. Reports should be generated that indicate users who have logged in during unusual hours or have exceeded their normal login duration. This includes flagging the use of the user's credentials from a computer other than computers on which the user generally works.	Configuration/Hygiene
CSC 16-14 (NEW)	Require multi-factor authentication for accounts that have access to sensitive data or systems. Multi-factor authentication can be achieved using Smart cards with certificates, One Time Password (OTP) tokens, or biometrics.	Advanced
CSC 16-15 (NEW)	For authenticated access to web services within an enterprise, ensure that account usernames and passwords are passed over an encrypted channel and associated password hash files are stored securely if a centralized service is not employed.	Advanced
CSC 16-16 (NEW)	Configure all systems to use encrypted channels for the transmission of passwords over a network.	Advanced
CSC 16-17 (NEW)	Verify that all password files are encrypted or hashed and that these files cannot be accessed without root or administrator privileges. Audit all access to password files in the system.	Advanced