Global Scientific JOURNALS

GSJ: Volume 11, Issue 3, March 2023, Online: ISSN 2320-9186
www.globalscientificjournal.com

SECURITY METHODS USED IN MALWARE (MALICIOUS) FILE AND DATA LOSES DETECTION

ALABI O. A., UGWU C. AND ONYEJEGBU L.N
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF PORT HARCOURT

Abstract
High performance computing is a growing industry of high-powered compute systems, built to calculate and run scripts most computer systems are incapable of, in run times fast enough to produce high-value and accurate data. This data is used by public and private sector industries worldwide. Given the high value of the output of these systems, this paper aims to review common security methods, attack purposes and types – and subsequently discuss how effective they will be in relation to HPC. Recent literature shows a growing number of attack vectors on traditional data centre systems, using a variety of methods and intentions of attackers. The efficacy of these methods differs when applied to a HPC instance – ranging from complete ineffectiveness to critical system failure, and high resultant cost. The recommendations involve a solution which apply relevant controls and security systems in an example implementation of HPC. An approach of layering reactive and proactive measures achieves a more secure environment, targeted at subsections of the HPC respectively. This considers the requirements of the HPC system to complete tasks with efficiency and without interruption and allows users the minimum required access.

1.0 Introduction

High Performance Computing (HPC) has been unlocking an ever-growing number of possibilities within research and industry since its first iterations in the 1960s. As the capabilities of HPC systems increase, so do the demands and possible applications for them, and while it is already heavily established in research and industry, there is still a huge amount of potential in new areas of study. The maturity of HPC is now at the point of increasing accessibility, with High Performance Computing as a Service (HPCaaS) being made available on more cloud computing providers. The reliance on the data produced for fields of study such as cancer research, natural disaster predictions and the military – it is apparent that the data produced will be of high value. The accelerated rate of progression seen within HPC has also brought down costs in computing. Weterstrand (2020) reports from the National Genome Institute that the cost of sequencing a single human genome has reduced from over $5000 to under $1000 since 2015 alone, which is a relatively small change compared to the cost in 2001 which stood at over $100,000. This makes projects and research feasible and accessible. Higher value data brings with it an increased likelihood of cyber espionage whenever the data is valuable to more than just the intended owner. Failing to implement adequate protection could lead to thief, intentional damage to the HPC, or destruction of data. Creating an impenetrable HPC is not, and will never be, an achievable goal.

Security applications, architectures, procedures, and policies focused the protection of the system and data are invaluable, but there will always be new ways to exploit systems and people.

Security in HPC is a unique problem, as these systems are generally designed with performance as the utmost priority, despite its output being so valuable. HPC also relies on its usage being optmised for the best cost benefit of the system, as such patching and downtime for maintenance is minimised as much as possible. The less frequently a system is patched and updated, the more vulnerabilities are available for a would-be attacker to use. Applications used within HPC are also constantly evolving and changing to meet new problems, and often contain bespoke code, which introduces application weakness.

The aim of this paper is to assess what steps can be taken to protect an implementation of HPC, while considering its unique architecture when compared to a traditional data centre, and highlight some challenges which HPC administrators should look to overcome. Security applications, policies and measures will be covered, as well as information around HPC design to overcome vulnerabilities.

HPC Structure

HPC itself is made up of a similar structure to that of a normal data centre. There is storage, compute nodes, occasionally some VM management and network switching and routing. The differences between the two environments appear when looking at how these factors are utilised, and how they are configured. HPC systems have a major focus on their ability to compute and complete tasks as quickly as possible, and as such need the ability for several nodes to work in parallel. HPC has several emerging architectures to adapt to new challenges.

Grid computing has been a way to achieve the sharing of resources in multiple locations to raise the capabilities of a single organisation. While this increases the capabilities of smaller soloed HPC instances, Mateescu et al., 2011, states that this is at the cost security. This is in part due to the fact data will need to be transmitted between sites, creating opportunities for data exfiltration.

Cloud computing presents the capabilities of HPC as a service, typically in a private or shared cloud, but it can be shared through a public cloud. This allows organisations with insufficient facilities to utlise HPC without needing investment in physical hardware and maintenance. Any sharing of tenancy also opens possibilities of data share to unwanted actors, which means this is an even weaker solution in terms of security. Regardless of the method used, the nature of HPC – running a strict path of tasks to achieve a result– makes it extremely predictable. HPC administrators are aware of what actions are being taken, and where data should be at any point in the process. Access is not needed beyond a certain point in this process, and as such permissions are kept simple and without administrative overhead. In contrast, a traditional data centre (DC) must be flexible enough to accommodate the multitude of systems on it and provide access to hosted services.

For these reasons, I will look to evaluate and discuss techniques used on traditional DCs and their appropriateness for use in single premises iterations of HPC environments.

Research Problem

While attack vectors, and methods open to would-be attackers, within HPC are minimal due to the nature of both its usage and the clients. This due to a few factors, the first being its inaccessibility, despite a rise in cloud solutions becoming available. Jonas et al., 2017, argues that cloud computing has a few inherent issues, including (but not limited to) the complexity of configuration between applications specific to the workload. This general problem of minimal accessibility and user base also makes it a smaller target for would be hackers or people of malicious intent, as it makes it a much harder job of finding out specific exploits and invasion techniques

to each HPC system specifically. The information provided by HPC systems are minimal by design and are generally a need-to-know basis. Finding out the available attack vectors will be informed by the architectures used by HPC.

HPC architectures are generally made up of three major components: compute, networking, and storage. Each of these components are of high importance to the system. The proposed study must first look to address how to appropriate it is secure each component separately, or as a group.

Aims and Objectives

The overarching aim is to identify and propose solutions for mitigating risks in data security specifically within HPC systems. A successful study will look to advise possible appropriate solutions or methods which better aligns the system to a more security conscious model. These proposed solutions, for reasons previously discussed, must take into consideration detrimental effects to the performance, integrity and condition of the data and system.

Proposed Research Methods

HPC systems are normally extremely expensive, so the ability to work on one for an extended amount of time, would be at extremely high cost. For this reason, the proposed study would be theoretical in nature, and as such, the research methods used will be informed by real-world data and architecture but will be classical in nature. Any proposed solutions will be largely untested and should be taken as advisory only until further studies clarify their suitability.

Terms and Abbreviations

Attack Vector (AV): Path used by a malicious code or unauthorised third party.

Command and Control (C&C): A signal from an infected machine to an attacker's server for instruction.

Data Centre (DC): The central computing infrastructure used by businesses for their compute and data storage, amongst other things.

Data Loss Prevention (DLP): Systems responsible for detective and preventing data exfiltration.

High Performance Computing (HPC): The capability to process high amounts of data in a short time, with the use of specialist techniques and application of computing hardware (HPC).

High Performance Computing as a Service (HPCaaS): This is an application of HPC which makes it available to users via a public or private cloud.

Internet of Things (IoT): The connection of real-world devices to the internet to monitor useful data.

Playbook: A sequence of predefined actions which can be triggered against an event.

Secure Shell (SSH): A protocol used which provides a secure command line interface over a network.

2.0 Literature Review

This review will identify common methods used to raise security within computing, so that they can be discussed against the unique challenges HPC presents.

a. Information Security Methods

This section will identify each security method, with an explanation of what it does, and the situations it is appropriate for, as well as possible impact on compute. It will be organised as one section per security method/system.

b. Anti-Malware

These solutions (commonly referred to as Ant-Virus) are mainly responsible for ensuring no malicious code is saved or run on a machine. Malware can be defined as any code which has intent to modify, extract or harm data and/or systems. There are a large variety of techniques and styles malware can employ. Amongst these, some of the more prevalent are as follows.

Traditionally, ant-malware solutions compare code found on a machine to a known database of previously discovered malicious code. This is available on two triggers – file access and on a schedule. File access scanning is typical of ant-malware solutions, and allows the scanner to analyse code before, or during runtime. Scheduled scans compliment this by scanning files at rest on a disk before they are accessed. While this is effective, the malware must be known to the solution, making it reliant on the intelligence being as up to date as possible. This does not defend against 0- day threats which were previously unknown.

Some ant-malware solutions have started looking towards behaviour analysis within machines. By investigating the actions that a file wants to perform before allowing it to do so – it can identify intent. This can be achieved by a process of sandboxing, which Shanmugapriya & Geetha, 2018, describe as 'a simple security concept; a sandbox is a "sealed" container, which allows untrusted programs to have executed within the sandbox.'

The nature of these methods makes the solution incredibly invasive – and often high impact on compute. Despite this, an end user may not observe any scanning take place on a typical computer. On large files, ant-malware may take substantial time and resources to analyse, which can be increasingly impactful on performance when done on-access.

c. Firewall

A firewall is placed on the transport layer at a junction , where one network interacts with another. This can be between an end machine and the network its connected to, or any intersection between networks above that. This allows control and monitoring of traffic and data along the transport path. This not only allows for a trust-based relationship between machines communicating, but also gives chance for data to be scanned for known malware or malicious behaviour as it passes through a firewall. Rules can be built up to control which traffic is allowed or denied by identifying trusted remote locations , applications, or ports and protocols.

Command and Control (C&C) traffic is a threat which firewalls are best placed to deal with. This is when a machine is infected with a small piece of code which simply asks the attackers server for further instruction and will perform the actions put on there. If left to communicate, the attacker has the potential to perform a coordinated attack against a network. These are commonly described as Bot Nets.

Denial of Service (DOS) attacks: are another method of attack in which an attacker will send traffic to halt service on a network. This is typically achieved either by flooding the connection point with more requests than it can handle, or by requesting a connection with a service that it then never completes – thus flling up available slots for connection.

Internet of Things (IoT) is a growing and integral part of consumer life, but a huge risk for our data. These devices often have no ability to install additional security. It connects personal data to a network, which could be removed – or the device exploited by attackers. The accessibility of these devices, and confidentiality of the data on them, relies on network layer security such as firewalls. These are pivotal in preventing abuse, and it is now even suggested that two firewalls should be put in place with different vendors to reduce likelihood of duplicated vulnerabilities (Nzabahimana, 2018). The nature of firewalls mean that they do not normally impact compute times but do (by design) restrict data flow in and out of systems.

d. Intrusion Detection

Intrusion Detection Systems (IDS) is an application that monitors activity and traffic for violations against policies. This can include user, system, and network activity and involve building up a portfolio of allowed activities by any of the parties. One

effective method of implementing this, is by allowing the IDS to record and define what is perceived as normal activity. Any activity which deviates from expected activity, or policy, can be fagged for action by an administrator, or a playbook ran against the suspected compromised entity. This workload can be hosted externally of a system and ingest from several different locations . IDS systems lend themselves to as many data sources as possible so that detection can become more intelligent.

IDS is now an established part of a security framework in large organisations, and within a security team. Further to this – 'analysis has shown that there is an increasing requirement for the development and training of anomaly-based HIDS solutions' (Čeponis & Goranin, 2018).

e. Authentication

At its core – authentication is the locked front door of any system its applied to. Its function is to ascertain the identity of the user, system, or process, and assign a pass which grants access to areas and resources which are applicable. Authentication is the core and first step in securing systems of any nature. An authentication platform typically has a process in which the third party is challenged, and a handshake method is used to grant an electronic pass, which details access levels and identity.

User authentication: is normally accomplished by means of a portal or prompt which quizzes a user for known information. A username and password is by far the most common authentication challenge, as this gives the system owner an opportunity to manage a common identity of the user through the username, and can communicate with the user effectively over their account as it is known to both parties. The username typically remains a constant throughout the use of a system. Having a password on each account allows for multiple benefits including variable complexity policies, and user-only knowledge of credentials. Complexity policies allow a system owner to define just how abnormal a password string needs to be to be accepted as a valid password. This is completed by setting requirements such as use of non-alphabetic characters, upper case characters and numbers. Generally, the more complex a password the harder it is for a hacker to exploit that password through password cracking. System owners can also set required password lengths, as well as the maximum age of a password which can prevent use of leaked credentials beyond the passwords expiry. Password files are stored encrypted and secure to a level in which administrators have no easy access to them, further reducing the chance of credentials being leaked and used maliciously.

While username and password authentication is commonplace for systems to use, it is an extremely exploitable method, with multiple vectors that hackers can use to gain access. Firstly, the credentials must be stored somewhere on the authentication platform, so loss of this file can lead to severe and mass loss of credentials. Encryption and access to this file is an extremely high priority in both the design of an authentication platform, and the administrators. As such – keeping minimal access to the file can be achieved using the least privilege principles, in that if an account does not explicitly need access to the file then it should not have the opportunity to do so. In instances where accounts do need access, it should only be granted if no indirect method is available. Another opportunity for exploitation in username and password is a human weakness. A study conducted by Komanduri et al., 2011, showed that the more a user perceived a password to be difficult, the more likely they were to write them down. Written down passwords are a clear opportunity for unintended actors to have credentials to a system they may not have access to and perform operations as a different user. For this reason, the National Cyber Security Centre (NCSC)

recommend several steps in keeping credentials safe, including adjusting policies to allow easy to remember passwords in the form of a passphrase, and supporting users in securely storing passwords (Password Policy: Updating Your Approach - NCSC.GOV.UK, 2018). Credentials form a "something they know" approach to security, and while this has many benefits – as soon as it is known by others then the security is almost ineffective and has failed to secure.

Another approach is to use something they have, a physical token which shows authenticity. When used in combination, a token with cover for the flaws of credentials. These are commonly in the form of a key fob or a dongle which use a known algorithm to create a new token at a short interval. This allows a system to expect a specific token at any given time, and only the assigned hardware token knows it. When the token is presented during the handshake, it allows the user to authenticate. Loss of a hardware token can be controlled as it can be made immediately invalid as soon as it is reported, mitigating the risks of unauthorised use. There is a reliance on the user both keeping the device safe and adhering to reporting procedures should it go missing, but when used in conjunction with a username and password combination – gives very little opportunity to get all necessary authentication requirements for any particular user. This combination is one of the common methods used to achieve a two-factor authentication requirement.

Recent development of multi-factor approaches has allowed users to use their own smartphones to provide a one-time password, or a human based authentication. This allows a user to register their own device, which increases the chance of the user knowing that their authenticating device is missing.

f. Data Loss Prevention

Solutions responsible for preventing and detecting loss of valuable data are commonly called Data Loss Prevention (DLP) systems. Policy driven systems can sit at different levels of the stack to make sure data at rest, or being transferred, is both authorised and being sent to the intended audience. These solutions utilise cooperation from other systems such as firewalling and antivirus to detect and respond appropriately.

Network based DLP solutions analyse traffic in movement to ascertain what is being sent where. When combined with known policies of what should be sent where – this can be effective in detecting and preventing data loss even when authentication has been bypassed. The effectiveness of the solution relies on how strict the policies are. For example, if sensitive data is identified that requires restriction on being sent to only certain destinations, the DLP solution can ensure this is the case – and fag up where it is not. Any access to the systems outside of accepted policy will be fagged, and with cooperation of networking infrastructure be blocked.

Client based DLP rely on endpoints having the software installed locally but have some benefits. Information gathered by endpoint DLP can be much richer, as it will be able to identify who the author of the data was, and what it was used for on each system. This can provide a useful timeline in the use and loss of data. Once again, policies drive what is and is not acceptable, and actions can be set against detections to determine what should happen once there has been a positive trigger.

Policies can only drive so much, and may be an administrative overhead, so a large amount of DLP solutions can detect sensitive information. There are many commonly identified information types which should be fagged as immediately sensitive such as passport numbers, credit card information and driving licenses. Having a system which can detect known sensitive information types within documents, can help in tagging, and control of the document from that point forward.

Larger solutions such as Microsoft Office 365 now have the DLP solution built into it, which also allows users to tag. Shown in the diagram figure 1, the solution proactively evaluates and re-evaluates documents for information which hits any defined policy. This allows for documents to dynamically adhere to policies when they are needed, with minimal user interaction. The cycle of steps are as follows – the DLP solution searches through all data for changes, and the search index is updated. When the data is privy to organisational policies, the DLP solution will poll the search index for the changes and take any resulting action.

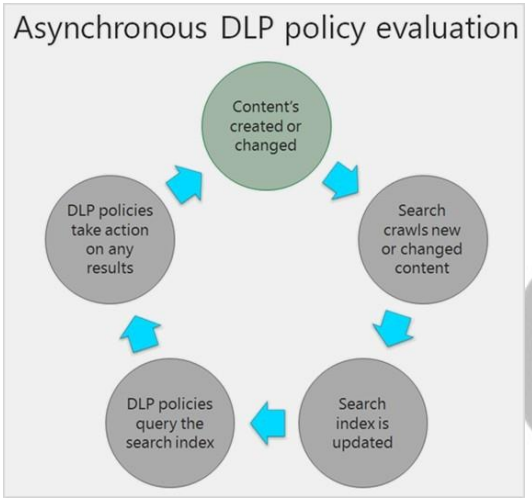

Figure 1: Asynchronous DLP policy Evaluation (Overview of Data Loss Prevention - Microsoft 365 Compliance Microsoft Docs, 2019)

Businesses can define their own sensitivity types within DLP solutions for automatic detection, commonly accomplished using regular expressions. Regular expressions allow for a format of a string to be detected, so the policy can be tailored to detect specific formats, such as usernames or file names – dictated by local naming procedures. When used in conjunction with communication methods within a business, control can be put on how data is moved around. Policies can dictate things such as automatic formatting of a document (such as a watermark) and set who documents may be sent to. DLP solutions can even enforce requirements such as Encryption on files of a certain sensitivity.

Monitoring

Monitoring concepts employ the logs collected by various systems to ascertain events that have happened, with reasoning where possible. Generally, the richer the dataset provided, the more useful it can be when reviewing for unexpected events. Auditing of core systems such as authentication platforms is commonplace, as it is used when troubleshooting a user's access as well as reviewing their access. Information stored in these events can be pivotal in forming a complete story of what has happened, and by whom. For example – if a user logs into a system in England, and then ten minutes later in Australia, it is easy for a security expert to fag this as an impossible travel, and review whether or not this account has been compromised, and the credentials known by actors besides the intended user.

More advanced monitoring systems can employ thumb-printing techniques to ascertain if the actor using credentials are in fact the user it belongs to. Techniques which build up an image of what normal activities, locations, and devices a user uses can build a picture of normal usage. When audit information is provided to a solution such as a security and information event management (SIEM). These SIEM solutions can harvest the data from audit logs and create thumbprints on accounts. This enables security professionals to build alerts and triggers for action which indicate suspicion or alarm, such as impossible travel of a user. SIEM solutions may also be able to employ playbooks to automate reactions to these unwanted events. The impossible travel event could be used to automatically send an email to the user (and possible victim), to prompt for clarification if this were them,

and disable the account where necessary to regain full control.

A SIEM solution can be extremely useful to security professionals when also dealing with known past events. Building up a full timeline of actions taken by a specific user can inform changes to policies to prevent future action. There are factors businesses may need to consider when writing audit policies. Retention periods can be vital in looking back through an event but may also have legal obligations in storage and removal. It should also be considered which events are audited and should be considered on a system by system basis. Auditing can take place not only on users, but devices and applications too. A business should consider all access, even non-person, when evaluating what they deem as necessary to be collected. Monitoring can be a powerful tool, and effective in detecting unwanted activities as they are happening and reflecting on why an event happened.

g. Code Review

Code review is the peer reviewing of written code. Human written code is inevitably going to include human error, so peer reviewing code addresses this and reinforces best practices, addresses bugs, and importantly identifies security weaknesses. It also has the added benefit of improving the quality of the software being produced, as inefficient code can be replaced where it is picked up. Generally, this is completed by team members within the same company, but some third-party code reviewing services are available. Reviewing code – from a security perspective – helps identify possible weaknesses in code, and instances where common exploitation such as cross-site scripting, or SQL injection can be used.

While code review is seen as standard practice, and has clear and unquestioned benefits, the practice is also not all encompassing. One study conducted by Edmundson et al, (2013) analysed the effectiveness of 30 developers reviewing provided code with common security weaknesses and found that none of the participants were able to identify all the flaws within the code. This clearly shows that the effectiveness of reviewing code needs to go through multiple stages of review for better diligence.

Manual code reviewing of course increases length of time and therefore cost of a project. One answer to this has been to automate it. The prevalence and development of automated code reviewing systems is growing and looks to address the weakness of peer reviewing code.

While code reviewing is often considered an important step in web and software design, it has a higher security benefit in instances where custom or bespoke code is running on systems by end users. In these instances, the risk factor of adverse effects of the code is much higher, whether it is poorly written code, or harbouring malicious code within it.

To effectively implement code reviewing practices, standards and procedures should be defined to set acceptable goals and requirements of written code.

h. Attack Purposes

Using the above literature review – this section aims to identify the range of purposes for an attack of any type and differ completely depending on factors such as the target, the incentive, and the intention of the attacker themselves. Some attacks do not have an intended specific target, and simply seek to cause impact as an achievement. The more effective and higher priority attacks come from those with specific intention, and have a set goal of destruction, data exfiltration, or data obfuscation. In this section, attack purposes will be identified and elaborated where possible. Identifying purposes in a system informs priority in protecting a specific system or data.

i. Destruction, Corruption and Encryption of Data

The action of destroying or modifying data to make it unusable would have clear consequences on the

ability of a system to perform its actions, and of an organisation to function correctly. The risk of severe damage is likely to be greater, the lower the stack the attack has access to. Data destruction at a high level such as file storage is likely to have little impact above the monetary value of the data destroyed, and the impact can be largely reduced using counter-measures such as frequent backing up of valuable data.

Destruction at lower layers in a compute or network stack can have many far-reaching implications and reduce not only the data contained on the system, but the ability of the systems and networks to function.

There has been a severe rise in attackers using the Encryption of data to extort money from organisations. In these cases, money is a clear incentive – and attackers will choose targets who rely on their data and can afford to pay for it. Organisations with poor backup policies or need for live data are especially vulnerable to these attacks. One survey carried out by Sophos (2018) stated that 54% of responding organisations have been a victim of ransomware over the previous year alone, and that most of them (77%) had up to date endpoint security at the time of the attack. This presents a clear opportunity for hackers to exploit and generate their own income at the expense of organisations.

j. Retrieval of Data

All data has some value, even when its purpose is to help a system run properly – such as a library or configuration file. When data has value outside of a system, it is a target for attackers to gain.

Data such as personal data has clear negative impacts on the privacy of the data subject, and the integrity of the data holder. Places to sell personal data are quite accessible on the dark web, and groups which look to exploit users can gain even more value out of this data. Credential information is a particular interest, as this allows an attacker potential to infiltrate the system while posing as an approved user – giving scope to more malicious and impacting actions. Identifying information such as contact details are used to exploit a person or target them for campaigns such as phishing. The introduction of the General Data Protection Regulation (GDPR) in the European Union looked to address and set baseline requirements of organisations holding data (data controllers), as well as grant rights to the subjects of the data.

Other data which have high value, such as results of research, and highly calculated data sets are also targets of thief. Bressler & Bressler (2015) notes that in 2007, $100 billion was the value of lost data due to business espionage, and companies spent $95 billion on cyber security systems to combat this. One example of this was Dongfan “Greg” Chung, a Chinese-born engineer who worked for Boeing in the United States, and reportedly over 30 years shared trade secrets with Chinese government for the personal gain of $3 million (Chinese-Born Spy Gets 15 Years in Prison - US News - Security | NBC News, 2010).

Insider threats make up a large portion of data removal. These can be in one of two categories – a turncoat, or an unknowing victim. Unknowing victims are people inside an organisation who are used and exploited by attackers to divulge information, such as their credentials, or being tricked into sending on information. These victims do not usually know they have done anything wrong and have no ill-intention towards the business when conducting these actions. The other category is turncoats, users who are inside and have a motve to export data from the business. Insider threats account for around 34% of data breaches (Verizon, 2019).

k. Malicious Intent towards the System

An attacker may simply seek to impede or stop the function of a system, to impact business or reduce the service. A great number of reasons why users do this exist, but some prevalent ones include; gaining an advantage over the businesses or even

opponents in gaming systems, removal of websites from public domain due to moral standings, or an insider with a motive (such as vengeance) to cause disruption.

People who enjoy playing games in their free time on a console or computer may have experienced this exact thing as a direct attack on them, and possibly not even known it. Fellow gamers play the role of cyber-attackers in degrading their opponent's network and computer performance during a game to gain an advantage. This is commonly carried out using a denial of service attack, where the attacker sends an unmanageable amount of data traffic over the internet to the targeted victim, slowing down their network, and therefore game performance. While most gamers can take simple steps to protect themselves, it may be unavoidable for most, and mostly relies on keeping your IP address unknown to third parties where possible, or use a VPN while gaming, according to Imperva (Protecting Gamers from DoS and DDoS Attacks, 2016).

Websites, businesses, and governments have all been targeted by users who did not agree with their practises and have had attacks on their system due to this. Hack activism is used to describe the use of technology to incite or change political or business agendas. One group which is collectively called Anonymous (Anonymous(@YourAnonNews)/Twiter) takes responsibility for the exposure of company, government, and group data of various companies where their idea of a just society is not being upheld. In one attack, the group took down and exposed the people behind the distribution of child pornography (Anonymous Shuts down Hidden Child Abuse Hub • The Register, 2011).

In one article written by Walters (2015), he states that 76% of companies surveyed were a victim of an attempted web-based attack. This huge figure shows the scale and ability to manoeuvre an attack at a much lower cost than that of the defence against it.

l. Unauthorised Use of Resources

Where the resources of a system are made available, it can be exploited – such as for cryptocurrency mining, or simply unauthorised use. Since the emergence of Bitcoin in 2009, cryptocurrencies have driven a surge in the use of computing systems to farm currency which can be monetarily converted. This has come hand in hand with crypto jacking – the means of an attacker using resources he does not have permission to use, to mine cryptocurrencies for themselves. Techniques to achieve this have also evolved and are moving away from being file based as they are too commonly detected my ant-malware solutions. File-less techniques, such as browser hijacking, allows the computer resources to be mined, without any malicious code being downloaded to the computer for detection. Carlin et al., (2020) notes that 'Such script-based malware attacks have increased in the past two years, with as many as 53% of breaches being initiated by normal ware attacks.'

m. Attack Types

In this section, I will look to discuss different attack types and their effective strengths and weaknesses. I will also look to link these to the different attack purposes, and techniques commonly used to combat these attacks.

a. Malware

A computer virus commonly modifies a legitimate executable or library file (an application dependency) to be run. This approach to infection makes them especially difficult to clean up, as it will be at the expense of a legitimate file, which could be necessary for software to run. (Grimes, 2019) goes so far as to state "The best antivirus programs struggle with doing it correctly", and simply opt for quarantining or deleting the infected file, instead of trying to reverse the changes made by the virus – a near impossible task.

Trojans are a type of malware which disguise themselves as legitimate software so that they can be

made available for unwilling users to download and run the code. Trojans are particularly prevalent on social media, and in places where fake adverts can be purchased, or pop ups can occur to trick the user. Once the malware is seated on the computer, it has a range of activities at it's potential, including creating a backdoor to the infected machine, downloading additional malware, acting as a fake program (normally ant-virus), or allowing remote access to an unauthorised third party user. One study into the propagation of trojans over social concludes that, among other necessities, 'AV products play an important role in protecting OSN users from Trojans.' (Faghani & Nugyen, 2017).

Worms are like viruses, but do not require the user to run them to spread further. These generally employ a known vulnerability of a system, especially at the transport layer, to propagate and infect systems. These can be particularly dangerous within organisations, as software updates are usually uniform across the estate, so if one machine gets infected because of a vulnerability – the rest are likely to follow, especially with an infected machine on the network.

Ransomware has a particular purpose, and a predictable function –in that it will look to encrypt data of value to extort currency from the targeted user. The sum demanded is normally requested through a hard to track system such as a cryptocurrency, and in 2016 was observed as being as large as 1077USD (Muhammad Arslan Tariq, 2019). Due to its low risk to the attacker, and high payoff, it is a growing threat.

Intrusion

An attacker's ability to gain access into systems allows them to conduct a high range of activities to their own choosing. They also have the strength of being able to see and react as things happen. If an attacker were to get into any system, and have an actual view of the data, the attacker can manually infect, encrypt, or manipulate files and systems. Attacks that involve intrusion techniques are normally by nature targeted, but there is no evidence to believe they have a specific goal when inside the system beyond access and disruption. Attackers are likely to make decisions on the most influential way to achieve this once they are inside a system, and past defences.

Intrusion attacks typically start with reconnaissance – the act of finding a weakness in a system to exploit. There is a plethora of available tools to achieve a picture of the best way to achieve access. Vulnerability scanners can be aimed at known IP addresses and ports and will scan against databases of known vulnerabilities to see if they can still be exploited and used. Attackers can also exploit users through means of credential gathering tactics, so that they have access to systems using authorised credentials.

Once a way in is found which fits the desired method of the attacker, they can use that exploit to elevate themselves to a position of access which they see benefit. Starting another task of information gathering is important at this point. If the attacker is looking to retrieve or manipulate data, and they already have access to data they deem valuable enough, they can then start the malicious and damaging phase of the attack. If their aim is to gain access to more information or shut down servers/services within an organisation – they will look to further elevate the access they do have. This can be done by repeating the same method of intelligence gathering, normally against the authentication system in place, then exploitation.

One high profile case involved two Iranian hackers, who managed to access and exfltrate hundreds or terabytes of personal and valuable information from servers all over Europe and the United States. This was a coordinated campaign of attacks spanning over years, in which victims include United States government, and used this data to sell and share with other third parties (The United States Department of

Justice, 2020). Intrusion detection systems are of course vital in countering attacks of this type. Knowing who is accessing what, and where data is coming from and should raise the security profile of systems.

Denial of Service

Attacks which look to disrupt a service through flooding are both simple and powerful. These attacks are commonly very cheap and simple to run and have a very small risk factor for the attacker, as they can be done in a variety of ways to cover their tracks. At its most basic, a denial of service attack looks to overload an open networking port or service by sending more traffic than the system can handle. The effects of this can be instant, and can range from a slow performance, to the complete unavailability of the service being offered. Attacks can be launched from a single computer or server or spread across multiple nodes to not only speed up the rate of disruption, but complicate efforts to combat the attack. The simple solution of blocking traffic from a single address is made much more difficult when the attacks are coming from multiple addresses. This is called a distributed denial of service attack.

As shown in figure 2, the attacker can scale up a denial of service attack with little to no further complexity as all nodes have one target, and one task to perform. The slaves shown in this diagram may not even be owned by the attacker and wait for instruction from master nodes. These master nodes, when coordinated, will instruct slave notes to initiate the attack on any given target.

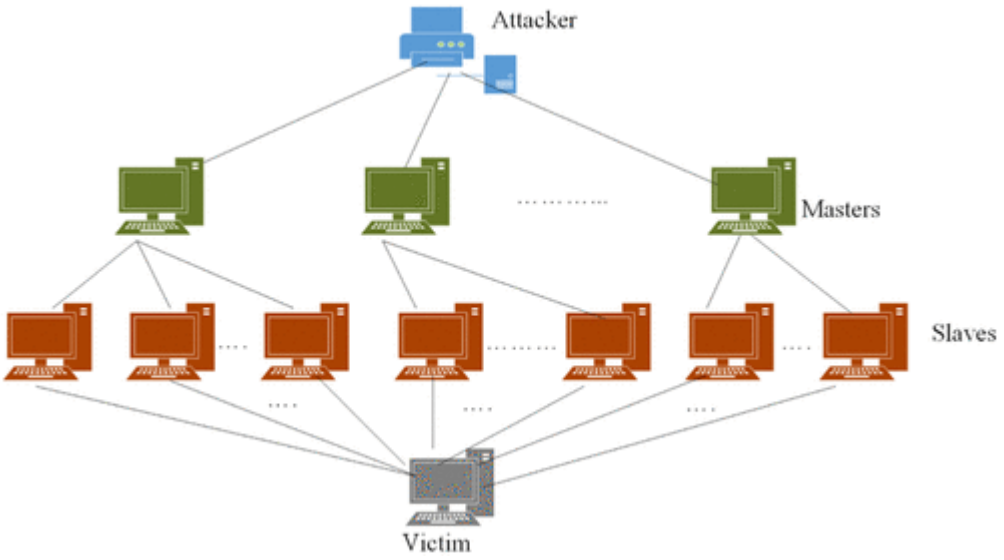

Figure 2. Structure of a DDOS attack (Mahjabin et al., 2017)

By using malware to compromise and weaponise unsuspecting user machines. The attacker then uses command and control traffic to have the compromised machines act as slaves and start attacking the victim on command. Defence against this can be even more complex, as these addresses can be vast in nature, and countermeasures can even restrict legitimate use of services.

Mahjabin et al., (2017) raise further concerns in the challenge against these attacks by stating ‘The increasing growth of the Internet and the availability of the insecure IoT devices are a very big threat of the current cyberworld. All recent major DDoS attacks are based on IoT botnets.’

Denial of service attacks are difficult in nature to respond to but can be defended against. Reactionary measures such as monitoring and

throttling of resources are a necessary step. Organisations are also engaging with cloud-based services to act as a gateway to their services, allowing for a much more resource available system to vet incoming traffic before the end service is affected.

n. Credential Gathering

It is a long-established core of information security practises that keeping credentials safe is one of the most important factors when setting up any system. Making sure user's passwords are safe, but also applying the least privilege principles make up the basics of an authentication platform, as these credentials are used to access the data which an organisation deems important enough to store. It is on this basis that hackers will look to obtain these credentials, as it would be the equivalent of walking through the door with the key.

The first, and most basic method a hacker can utilise to obtain credentials is through password cracking. In its simplest form, this is automated guessing of passwords against a dictionary or algorithm. The program will continue to try passwords until it has been successful in gaining access to a system and will then alert the attacker to the known credentials for their use. This basic password cracking technique can be easily defended against, by implementing controls on both passwords and password attempts. When complex passwords are enforced by a system, the time it takes to crack these becomes exponential, but introduces new issues. Loss of passwords through human mistake is much more likely when passwords are hard to remember and are likely to be written down or stored inappropriately. Kävrestad et al., (2020) found that 'using phrases with four or more words as passwords will generate passwords that are easy to remember and hard to attack.' A password styled like this is both easy to construct and remember by a user, but also incredibly unlikely to be guessed or cracked, even by automated solutions.

Another tool an attacker can also use is social engineering – tricking a user into handing over information and credentials. Commonly this is deployed through a phishing attack, urging a user to visit a website which appears legitimate and asks for their credentials. Phishing attacks can prey on the vulnerable in an organisation, especially by acting as a senior and causing urgency in completion of a fictional task.

Important but simple steps can be taken to both reduce the likelihood of compromised credentials, but also detect and react to access when they are. Profiling of users can be useful in determining whether the intended user is in control of an account or not. If the accountant has logged on for the past two years from only two locations in England, and then tries to login from the United States – it's an obvious red fag that it may not be legitimate access, and reactions to this impossible travel can be automated. Profiling can also be done on a session level – looking at information passed during the connection to a system to ascertain what a user's regular session looks like.

Biometrics are also used to identify a person as l above know information such as credentials, though it has flaws. As humans our features change due to all sorts of reasons; environmental, dietary, style choices, injury, and ageing. There are also questions regarding their acceptance, as the data to verify the biometrics of a user is incredibly personal, and some view as invasive. Research shows that there are differences culturally with regards to the trust and reaction to the use of biometrics (Riley et al., 2009), with variable levels of acceptance. With these caveats, there are applications where biometrics can be used effectively to ensure proper access, and devices such as fingerprint readers can be systematically deployed as an effective defence mechanism.

Separating out credentials over multiple accounts of different purposes mitgates the damage of

compromised passwords. The less systems, services, or data a credential set has access to, the less can be accessed by an attacker with them. It is commonplace for organisations to separate out higher level privileges with different accounts, so that they can ensure accounts which have a higher value, can have higher levels of security such as monitoring, stricter password policies, and multifactor authentication .

o. Vulnerability Exploitation

Platforms of access available to end users such as websites, applications, servers, and desktops, all provide an opportunity for exploitation. Organisations which provide these platforms benefit by updating and patching systems, and not using systems which are no longer updated by their publisher. Vulnerabilities can be discovered by both would-be attackers, developers, and end users. Reporting and confidentiality of these vulnerabilities is paramount if the issue is to be addressed, and the reaction of the developers in deploying both an update, and where necessary a workaround, is vital in maintaining system security.

p. Internal Attacks

As previously mentioned, the reality of internal attackers is a very real one, and an extremely dangerous one. Internal attacks do not necessarily rely on bypassing technical controls, making detecting and reacting to them extremely difficult. Data loss Prevention solutions offer one of the few measures against internal attacks, as we can use them to ensure data is being used for solely the purpose it was designed and transported in line with trusted sources. The weakness of these solutions is in the need for the end user experience to be constructive in being able to work with and produce data. Internet access is almost a necessity for most organisations, as resources and cloud compute are more readily available. This introduces unknown risk of users downloading and executing malicious code, whether intentionally or not.

3.0 Professional Issues

As this paper looks to address security concepts and attack mitigations within HPC environments, it should not be used as a term of reference to assess the security posture of a live system. The intention of this paper is not to call attention to, or back up any claims of weak security within a live HPC system, especially where a contract of use exists. While some HPC knowledge and research was done within HPC professional groups and events, this paper also does not mirror or use the specific setup of any HPC instance known. The design proposed is a theoretical one and is used to demonstrate a possible approach to security concepts within HPC, without commenting on structure or use of PC system design.

This paper also has no specific HPC instance as a target for development, so should not be used explicitly to inform addition of security concepts or technology, but as a basis for discussion and research around possible relevant controls.

HPC Considerations

When looking at raising the security posture, there are considerations specific to HPC which may change both the effectiveness of the security mechanic, and the value of the system itself. This section will discuss considerations when looking at security solutions and will be used to reflect on in the following sections.

HPC's value is derived from its ability to handle extremely large data sets as quickly as possible. Since 1993, the TOP500.org have been tracking and ranking supercomputing systems worldwide. One reason for this is so that potential clients can see in a clear list format just how fast the system is in comparison to competitors. Within this time, the website shows a huge shift in potential of systems to be able to process data at extremely high speeds. In 1993, the website reports the fastest supercomputer running at a theoretical peak performance of 131GFlops/s, to 200,794.9TFlops/s in November

2019 (Top500.org), which is one and a half million times faster. This clearly shows the immense drive to process data as fast as possible, with hardware designed specifically for HPC purposes.

The second consideration is in the scalability of HPC systems. Particularly in cloud and grid HPC instances, the ability to add and remove nodes allows for a transformative approach to meet the needs of any running job. These systems need to ensure they can add nodes into a system without being impeded by security. Given their reliance on the transport of data over distanced networks, they are also required to make sure that the data delivered to them is timely and unchanged.

Data interacting with HPC has a potential to be extremely high value. As the internet of things expands, there is an ever-increasing demand to be able to process huge data sets to gain insight, especially into areas which directly impact income for businesses such as marketing. HPC is also used to predict and automate stock trading in the finance industry, and high workload artificial intelligence. It is for this reason that the security of the data itself should be a high consideration in any security implementation.

As an overlooking design, as shown in figure 3, HPC is a simple concept – an input and scheduling system, a compute section, and some output disks. For users, there is a single point of entry, and a single point for retrieval of data.

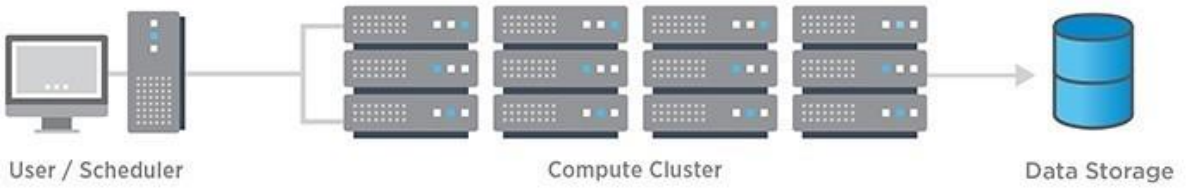

Figure 3: HPC 3 main components (NetApp)

This allows HPC administrators to control and restrict access to the system without complexities that a normal data centre where multiple users presented services have different ports and entry requirements. Access to HPC is usually via secure shell (SSH), which provides a command line interface to the user, enabling them to manage and utilise the service. This single route and interface mean permissions and access can be granular. The single platform for entry also eliminates any complexities in monitoring, as the only traffic should be coming through a single protocol.

Another complexity of HPC is in the data storage. Given how vast in size the data sets being put into HPC can be, an emphasis is placed on protecting that data not only at rest but in transport too. Though the raw input data is inherently less valuable than the resultant output data, both sets are likely to be large, and thus a potential challenge when keeping secure.

User access is also a point of consideration, as generally all users on the system will have been privy to either extensive experience or training before being granted access. Effective use of HPC requires users to understand and modify code and parameters for any one task. While this is not beneficial for general accessibility and ease of use – it does imply a skill level sufficient to operate. It is unlikely that a user with no knowledge or training in a particular HPC instance would have the ability to successfully start and run a job. This means that HPC administrators can expect predictable actions from users, and deviation s from this may indicate

system abuse. User training also reduces the likelihood of system or task failure due to human error, shown in a study by Schroeder & Gibson (2010) in figure 4.

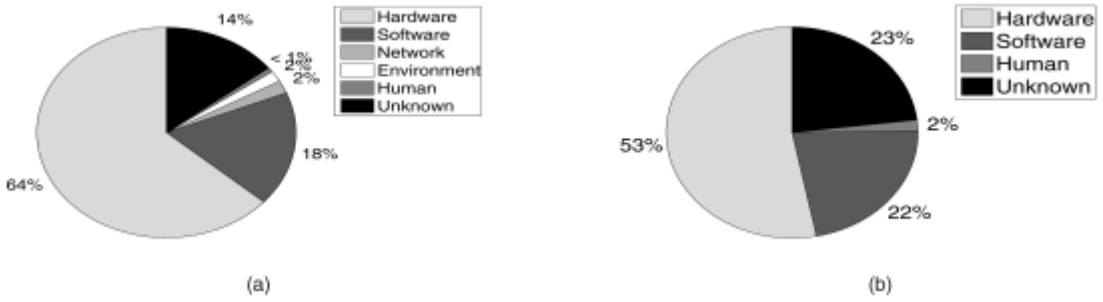

Figure 4: The breakdown of failures into root causes for the LANL systems (a) and system X (b). The LANL graph shows the breakdown across all systems (A-H) (Schroeder & Gibson 2010)

What is also shown here is the prevalence of failure within the hardware. This is unsurprising since the hardware is pushed to its functional limits where possible to provide more value to the system. Failures are normally accounted for, and tasks will assume the possibility of hardware failure for failover purposes. This is a huge beneficial reaction to the functioning of the system, but may open the system up for abuse if uncontrolled, and will rely on task control to ensure compliancy.

One final consideration is the workload model of HPC solutions allowing for fail over and contingency. By design, tasks are scheduled, and nodes are allocated to tasks through the scheduling system. Tasks can last days on HPC, with little disruption to the task beyond small IO interruption s to catch snapshots as a backup. If a node experiences a failure, it is assumed a HPC instance can reassign and restart the workload from another node.

Security for HPC

This section of the paper looks to address the reviewed attacks types and purposes with a view to identify the level of threat it poses to a standard HPC system, and mitigation techniques will be categorised into reactive, active, and proactive measures. This will be followed by implementing these ideals into practise as a speculative high level HPC design with recommendations on approaches to addressing them with the HPC consideration s. The design and recommendation s are purely theoretical. Because of the vast differences between instances of HPC, assumption s are made, and the design is with focus on a single premises system.

Data Loss, Encryption, or Destruction of HPC Data

While this covers several attack vectors, this specific area is a high priority for HPC systems. The service provided by a HPC system is solely the ingest and production of valuable data. There are several points which need protecting to cover the full scope of valuable data. The earliest point is the transfer of raw data to a HPC system. While this data may be more difficult for an attacker to capture, given that it is normally very large in size, it is still valuable data – and therefore a potential target for an outsider. Protocols such as SSH are already commonly used for login to a system, and where block file storage is presented externally, protocols such as SSH File Transfer Protocol or File Transfer Protocol Secure. SSH File Transfer Protocol is an addition to the

already commonly used SSH protocol and allows for secure transfer of data across a network and has been developed for use with cryptographic protocols such as Transport Layer Security. Implementation of this would protect the data during transport, and if intercepted provide a level of confidence that the data is unusable in the captured format. File Transport Protocol Secure also looks to meet the same demand, and uses the common File Transfer Protocol, with an implementation of Transport Layer Security to protect the data within. Either of these two methods would provide a level of security for the data during transfer to the system, but do not make up an extensive list of methods to do so. In instances where HPC is located at a specific site, hard copies of data brought to the site to eliminate risk of data loss electronically – especially in places where the data itself has a high confidentiality rating, or would have National or inter-national level implication s if lost.

The next stage that data could be affected is data uploaded for ingest by the system. Data within HPC is stored in a location accessible to the scheduler, or master node of a HPC instance. Depending on the imminence of use, the data itself should be encrypted and stored in a location with access by only the system processes and necessary technical staff, with a view to ensure that no access outside of what is required takes place.

Insider threats within HPC is already mostly designed out. If permissions to data storage are restricted by the user, there is no scope for end users to divulge data other than that which they uploaded anyway. Complete auditing of user actions, especially against accounts with higher privileges such as technical staff, will ensure awareness of suspicious actions and may indicate insider loss of data. One other consideration to mitigate the threat of an insider incident is over the devices which connect to the system itself. Basic posture checking over end devices should be set at a security level appropriate for the system. Posture checking grants the HPC system owners an opportunity to set minimum levels of security for any device attaching to the network, including presence of ant-malware, the running of an ant-malware scan, a check to see how updated the ant-malware system is, and checking patch levels of the operai system itself. Once a device passes these posture checks, usually directly before or after the authentication process, it can be passed through to the system. If a device does not meet posture check requirements, the system owners have an opportunity to deny access to that device, as well as fag up concerns over the user themselves, and act accordingly with measures such as invalidating the credentials. Further to this, network access to the system should be segregated where possible, and appropriate network traffic restriction s be implemented with the use of networked security devices such as firewalls, in effort to prevent addition al insider attacker access. With these mitigation s in mind, the HPC systems should be capable of ensuring minimum risk of insider threats.

Malware scanning on files will greatly reduce the chance of data destruction, but implementation of them may be contentious for the system. With a view to minimise the invasiveness of security on the compute section of HPC, malware scanning could take place as the data is ingested. This gives opportunity for a full and thorough scan of programs and raw data intended to be ran, without impacting active performance. Given the bespoke nature of programs run against HPC, there is a chance of false detections. For this reason, reactions of malware scans should not look to delete or remove data or code. By design, if the ingested data is clinically only available to be modified by the compute node itself, there should be little benefit in scanning the data on access, or on nodes using the data. Efforts made to clear data as safe well before the task is scheduled to run will need to be a consideration

when uploading, especially as the time it takes to scan a file is directly correlated with the size of the file itself, and the term big data is commonly used within HPC and speaks for itself.

Vulnerabilities of the system itself should take a lead in security consideration s, especially at the point which it is exposed. The protocol of connection used (commonly SSH) should be kept up to date, and older versions explicitly removed from the capabilities of the system, as this could lead to unwanted access to a system, with a potentially high criticality if exploited. Management of patching and vulnerability scoping, along with regular penetration tests ran against the environment, can gives as much assurance as possible that know vulnerabilities cannot be exploited to gain access to a system.

Given the implication s of code being directly intended for specific datasets, completeness and authenticity of the code should be upheld to ensure destruction of data will not occur. This can be carried out via signing and peer reviewing of code, and complete version history control over the code. Depending on the value of the data, the appropriateness of security surrounding the code should be considered, as malicious changes to code would have a monetary impact, especially if unnoticed before it is running. As described before, peer reviewing will be essential to ensure efficiencies are in place and prevent bugs which may reduce the efficiency (and therefore cost more), or stop the task being ran. Certification and integrity checking code can also embed assurances that code is unaltered and trusted.

Attacks on HPC Systems

HPC systems themselves are a likely target for attackers for as many motivation s as there are application s. Where HPC is used to conduct important research by any company, industry, or nation – it may become the focus of an attack to prevent its value. By design, single-premises HPC systems have very little opportunity for outsider attack, given its few and inflexible paths of entry – the likelihood of common attack vectors being relevant within HPC are minimal. HPC systems should look to contain this as a fact and ensure traffic from outside is controlled and protected. Distributed denial attacks and vulnerability scanning will inevitably show or create weaknesses in a HPC systems. Firewalling can help control access not only through proper routes, but from trusted locations . External network protection can alleviate malicious network loads and prevent denial of service attacks.

Secure HPC Design

A high-level concept diagram of a theoretical solution is shown in appendix b. The following section aims to describe processes contained within this. The solution itself is split up into three different areas – with different consideration s for each.

The first area is user access and ingress of data, which involves access to both the system and the data storage for the end user. One important point to note in the first stage is the necessity of scanning and integrity checking of users, files, and processes. This is the section of the solution with the highest risk against the system, as it allows for data to be written, potential for events which trigger misuse of data or resources, and potential exploitation of job schedules.

This is followed by the compute section, which is the critical part of a HPC system in terms of its value. Security measures in this section are shallow as to not impede this function, but risks are removed via structure design. Access to this part of the system will largely be direct and by authorised HPC technical staff, and therefore physical building security will be paramount in maintaining the integrity of the system. The compute section is surrounded by firewalling to control access into this section, and should be restricted by application ,

port, and user, with a default rule of deny and exception s built up only where explicitly necessary. The final section is user access and egress, which will typically be the service for users to retrieve resultant data. Access to this data should be read-only by default, and authentication ideally would be separate from that assigned to the ingress section, reducing the impact of improper use of an account.

Reactive Security

The first and most simple step is to collect logs of actions and events within the HPC system. Several different event types should be considered a priority - successful and unsuccessful login attempts, changes to credentials, and access to restricted areas. These simple logs will form an important picture on the investigation of any incident within the system. Retention of these logs should be long enough to enable a historic view of an incident discovered some time after it is discovery. The retention time span will rely on the availability of storage and systems to make the logs useful to a technical operator or security team member.

Data access, modification and transport would also be used in a complete HPC security design. This information ted into access logs will inform legal requirements of reporting a breach to users of the system, legal representatives, and local law enforcement where necessary. As with log management, retention of these details should be considered against their potential usage in a security incident.

Patch management is also an essential role in keeping both the perimeter and inside components of HPC as free from technical flaws and exploits as possible. Updates should be deployed as soon as there is faith in its completeness and effectiveness – and where possible should be testing in a development or research environment first. Once confirmation of the effectiveness of a patch has been made, the updates should be deployed as soon as is fit for the usage of the system. HPC has a detriment in that nodes and systems may be used extensively and uninterrupted for days at a time – and stopping a running job to patch the system may lead to loss of efficiency and income, as well as corruption or damage to the job itself.

Any data disk used for storage, especially that available to an outside user, should be privy to a full anti-malware scan where possible, with particular emphasis on the scripts and programs due to be ran on the compute nodes of the system.

Active Security

On access malware scanning where possible should be applied in both the ingress and egress sections of the solution. This may cause a level of impact in write times to the disk for raw data. Given the data is large, it may be assumed that this impact will cause a considerable difference in time to upload to the system. Another point to note here is that data detections found should be isolated where possible, but not deleted, to prevent false positives corrupting or destroying data structures of valuable data.

Authentication systems are a paramount part of this design, and separated into three categories – ingress accounts, egress accounts and administrative accounts. Ingress accounts should only have access to necessary SSH interface for the scheduler, and disk access to upload both raw data and software to be ran on the system. Egress accounts will have read-only access to the specific resultant output assigned to it. Administrative accounts are likely to have higher privileges of access, but addition al steps to further security can be taken by assigning role-based admin accounts, further reducing compromised account impact.

On all accounts on the system, multi-factor authentication is a necessity, and should be set as strict as the environment allows. These policies should be informed by access patterns of regular users, and acceptable limits. This can include an authentication app which provides a one-time

authentication password or token to the user, which will only be valid for duration dictated by policy. Another alternative is having a user entered phone number which is specific to them which can be texted for the same purpose. Having this control in place along-side a traditional credential set of a username and password will near ensure use of an account by anyone other than the intended person is near eliminated, if the system itself is protected. User condition s of access should dictate the reporting of loss of an authentication method, which should be made invalid as soon as possible. Scavenging of unused accounts will also further remove risk of unwanted access. Where viable, permissions should also be set to the necessary storage areas only, and segmentation of these areas should be as thorough as possible to reduce risk of data leak.

Firewalling between systems and data is essential in controlling network traffic. Where there is no need for two systems to interact, it should not be possible. This is with a particular view to the edges of the compute section, which includes the disk for both compute read and resultant write. This should be reinforced with segmentation of networks, as to not rely on the capabilities of a firewall. Should malware or a vulnerability aware attack propagate on a system, this sets clear boundaries of access and infection.

The compute scheduling agent should not only assign, but limit both resource and node usage on a task, to prevent the risk of malformed code causing abnormal usage. One opposing factor of this may be in the ability to predict and throttle the desired performance of the system around the task itself.

Proactive Security

Using a sandbox appliance, it may be feasible to execute a program or script against a subset of data to monitor actions during runtime. This approach identifies unknown malware and can inform administrators regarding the intended use of the system.

Anti-malware solutions should also prevent execution or access of files other than what is intended. By blocking unacceptable behaviour, such as the encrypting or renaming of files, or the creation of new files, will also prevent unknown attacks and malware from successfully executing within a system.

A testing schedule should be implemented and carried out, as this will help inform how successfully a system can recover from an attack. This should include, but not be limited to, recovering specific data from backup, the loss of a node, or the loss of a service dependency. Having this plan looks to address the goals of having a verified working disaster scenario response plan.

Vulnerability scanning throughout the systems, and penetration testing from the outer perimeter will identify weaknesses in the system before they become a live incident. Where possible, penetration tests should be done by a tester who is external to the organisation or system. This will act as simulation s for the possibilities of an outsider attack and will allow for a full review of the security procedures and systems in place.

Data while not in use by the compute section should be encrypted. This will prevent use of the data should data leak occur. Having the data encrypted ahead of possible access to it ensures that should it fall into the wrong hands, there is a level of assurances that the data is inaccessible.

A managed operating centre, and dedicated staff should be employed to monitor the security solutions in place. In conjunction with system resource data, this allows for security experts to have first-hand access to the operations and ensures the best possible protection of the assets.

Security training for all users embeds good security practise, and adherence to system usage policies. There are factors of information security

which rely on the user cooperation with system administrators, knowledge of proper usage and how to report deviances from set policies, and diligence in maintaining security access granted to them. Training should precede access to the system and should be reinforced whenever the relevancy of the training changes, and on a schedule such as yearly.

4.0 Conclusion

High powered computing systems by design do not engage well with security factors which slow them down, as any impact to performance also impacts its value. Effective security must balance with functioning of a system, otherwise it also devalues any reason to have it in place. There are numerous security services, systems and procedures that can be put in to raise the security landscape of a HPC system and reduce known attack vectors.

There are further studies which could be explored to identify specifics into how HPC systems can be secured. The impact of more invasive controls on compute, such as behaviour analysis, would provide an insight and open the possibility of new solutions bespoke to HPC. Human factors such as usability when addition al controls are added may also impact the efficacy of security controls. Research may also be informed by the gathering of currently used security controls in industry, and their effectiveness in real life scenarios.

In a traditional data centre and its presented services, there is an acceptable level of service degradation (though normally extremely minimal) that allow for heightened security measures – with a view to maintain value and privacy of data within. While this is not explicitly the same within HPC, this paper details methods which may be effective in protecting the systems and data.

5.0 References

Akusok, A., Bjork, K., Miche, Y. and Lendasse, A. (2015). High-Performance Extreme Learning Machines: A Complete Toolbox for Big Data Application s. IEEE Access, 3, pp.1011-1025.

Anonymous shuts down hidden child abuse hub • The Register. (2011, October 24). https://www.theregister.com/2011/10/24/anonymous_fght_child_abuse_network/

Archer.ac.uk. (2019). ARCHER ». [online] Available at: http://www.archer.ac.uk/ [Accessed 28 Apr.2019].

Bressler, M. S., & Bressler, L. (2015). PROTECTING YOUR COMPANY'S INTELLECTUAL PROPERTY ASSETS FROM CYBER-ESPIONAGE. 18(1), 15.

Brook, J. (2019). How Secure is High Performance Computing.

BSC-CNS. (2019). BSC-CNS. [online] Available at: https://www.bsc.es/ [Accessed 02 Apr. 2019].

Carlin, D., Burgess, J., O'Kane, P., & Sezer, S. (2020). You Could Be Mine(d): The Rise of Cryptojacking. IEEE Security Privacy, 18(2), 16–22. https://doi.org/10.1109/MSEC.2019.2920585

Čeponis, D., & Goranin, N. (2018). Towards a Robust Method of Dataset Generation of Malicious Activity for Anomaly-Based HIDS Training and Presentation of AWSCTD Dataset. Baltc Journal of Modern Computing, 6(3). https://doi.org/10.22364/bjmc.2018.6.3.01

Chinese-born spy gets 15 years in prison—US news—Security | NBC News. (8th February 2010). Retrieved 20 July 2020, from http://www.nbcnews.com/id/35300466/ns/us_news-security/t/chinese-born-engineer-gets-years- spying/#.X2d4Q2hKiHs

Edmundson, A., Holtkamp, B., Rivera, E., Finifer, M., Metler, A., & Wagner, D. (2013). An Empirical Study on the Effectiveness of Security Code Review. In J. Jürjens, B. Livshits, & R. Scandariato (Eds.), Engineering Secure Sofware and Systems (Vol. 7781, pp. 197–212). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-36563-8_14

Faghani, M. R., & Nugyen, U. T. (2017). Modeling the Propagation of Trojan Malware in Online Social Networks. ArXiv:1708.00969 [Cs]. http://arxiv.org/abs/1708.00969

Grimes, R. A. (2019, May 1). 9 types of malware and how to recognize them. CSO Online. https://www.csoonline.com/artcle/2615925/security-your-quick-guide-to-malware-types.html

Hashem, I., Yaqoob, I., Anuar, N., Mokhtar, S., Gani, A. and Ullah Khan, S. (2015). The rise of "big data" on cloud computing: Review and open research issues. Information Systems, 47, pp.98-115.

Hpc-sig.org.uk. (2019). HPC-SIG – The High Performance Computing Special Interest Group for UK Academia. [online] Available at: https://hpc-sig.org.uk/ [Accessed 02 Apr. 2019].

Jonas, E., Pu, Q., Venkataraman, S., Stoica, I. and Recht, B. (2017). Occupy the cloud. Proceedings of the 2017 Symposium on Cloud Computing -SoCC '17.

Kävrestad, J., Lennartsson, M., Birath, M., & Nohlberg, M. (2020). Constructing secure and memorable passwords. Information & Computer Security, ahead-of-print(ahead-of-print). https://doi.org/10.1108/ICS-07-2019-0077

Komanduri, S., Shay, R., Kelley, P. G., Mazurek, M. L., Bauer, L., Christn, N., Cranor, L. F., & Egelman, S. (2011). Of passwords and people: Measuring the efect of password-composition policies. Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI '11, 2595. https://doi.org/10.1145/1978942.1979321

Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W. (2017). A survey of distributed denial-of-service attack, Prevention , and mitgation techniques. InterNational Journal of Distributed Sensor Networks, 13(12), 155014771774146. https://doi.org/10.1177/1550147717741463

Mateescu, G., Gentzsch, W., & Ribbens, C. J. (2011). Hybrid Computing—Where HPC meets grid and Cloud Computing. Future Generation Computer Systems, 27(5), 440–453. https://doi.org/10.1016/j.future.2010.11.003

Met Ofce. (2019). Weather data for business - Met Ofce. [online] Available at: https://www.metofce.gov.uk/services/data/business-data [Accessed 02 Apr. 2019].

Muhammad Arslan Tariq Muhammad Arslan Tariq. (2019). The Evolution & Growth of Ransomware: IJECI, 3(1), 5–5.

NetApp. (n.d.). What Is High-Performance Computing (HPC)? | How It Works | NetApp. Retrieved 21 July 2020, from https://www.netapp.com/us/info/what-is-high-performance-computing.aspx

Nzabahimana, J. P. (2018). Analysis of security and privacy challenges in Internet of Things. 2018 IEEE 9th InterNational Conference on Dependable Systems, Services and Technologies (DESSERT), 175– 178. https://doi.org/10.1109/DESSERT.2018.8409122

Overview of data loss Prevention —Microsof 365 Compliance | Microsof Docs. (2019). Retrieved 8 August 2020, from

https://docs.microsof.com/en-us/microsof-365/compliance/data-loss-Prevention - policies?view=o365-worldwide#how-dlp-policies-work

Password policy: Updating your approach—NCSC.GOV.UK. (2018, N https://www.ncsc.gov.uk/collection /passwords/updating-your-approach November 19).

Protecting Gamers from DoS and DDoS Attacks. https://www.imperva.com/blog/protecting-gamers-from-dos-ddos-attacks/ (2016, June 22).

Riley, C., Buckner, K., Johnson, G., & Benyon, D. (2009). Culture & biometrics: Regional differences in the perception of biometric authentication technologies. AI & SOCIETY, 24(3), 295–306. https://doi.org/10.1007/s00146-009-0218-1

Schroeder, B., & Gibson, G. A. (2010). A Large-Scale Study of Failures in High-Performance Computing Systems. IEEE Transactions on Dependable and Secure Computing, 7(4), 337–350. https://doi.org/10.1109/TDSC.2009.4

Shanmugapriya, K., & Geetha, C. (2018). PATTERNS FOR IT SANDBOX INNOVATION. InterNational Journal of Pure and Applied Mathematcs, 119(12), 12. Sophos. (2018). The State of Endpoint Security Today.

The United States Department of Justce. (2020, September 16). Two Iranian National s Charged in Cyber Thef Campaign Targeting Computer Systems in United States, Europe, and the Middle East. https://www.justce.gov/opa/pr/two-iranian-National s-charged-cyber-thef-campaign-targeting- computer-systems-united-states

TOP500. (n.d.). Introduction and Objectves | TOP500. Retrieved 21 July 2020, from https://www.top500.org/project/introduction/

Verizon. (2019). 2019 Data Breach Investgation s Report. Verizon. https://enterprise.verizon.com/resources/reports/2019-data-breach-investgation s-report.pdf

Walters, R. (2015). Cyber Attacks on U.S. Companies Since November 2014. The Heritage Foundation Issue Brief, 4487.

Weterstrand, K. (2020, August 25). The Cost of Sequencing a Human Genome. Genome Research Insttute. https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost