An NCC Group Publication

Peeling back the layers on defence in depth…knowing your onions

Five security measures that'll frustrate the bad guys…and maybe the security consultants too

Prepared by:

Ed Williams & Grant Dale

Contents

Document History

Issue No.	Issue Date	Change Description
1.0	14/10/2016	Approved for public release

1 Introduction

NCC Group is a leading provider of professional information, security testing, red teaming, phishing and other forms of real-world attack simulation. Our extensive cyber incident and defence operations experience has resulted in us gaining significant insight into the techniques and methodologies used to breach organisations of all types and as such, we are uniquely positioned to advise on how to best frustrate, deter and discourage threat actors when they attempt to gain access to sensitive resources.

It seems not a week goes by without news of a cyber attack making the headlines,[1, 2, 3 & 4], so it is imperative that organisations are fully prepared for malicious attacks from both motivated external attackers and internal threat actors. To help combat such issues, CESG has produced good high-level strategy[5] document and ten steps to reduce cyber risk. This whitepaper will go a step further and discuss five key principles of network design and implementation that, when combined, create the foundations of a defence-in-depth strategy that will provide an organisation with increased assurance, reduce the impact of breaches and ultimately frustrate any malicious threat actors that do breach the perimeter.

The principles discussed in this whitepaper are not new but they are all too often ignored or only partially implemented and are therefore not providing the necessary levels of assurance.

[1] https://en.wikipedia.org/wiki/Ashley_Madison_data_breach
[2] https://en.wikipedia.org/wiki/Sony_Pictures_Entertainment_hack
[3] https://www.troyhunt.com/the-dropbox-hack-is-real/
[4] http://money.cnn.com/2016/09/22/technology/yahoo-data-breach/
[5] https://www.cesg.gov.uk/10-steps-cyber-security

2 Defence-in-Depth

So how do we frustrate threat actors? The basic, high-level premise is defence-in-depth[6], whereby the undermining of a single component, through whatever means (including zero day attacks[7]), does not fully compromise the intended target. Instead, multiple individual compromises should be necessary to achieve a full compromise. For example (and this does happen), accessing the Internet from Internet Explorer 8 on an unpatched domain controller. While this is clearly a worst case scenario, it offers no defence-in-depth or resilience against threat actors.

The following graphic demonstrates the individual components that ultimately create a defence-in-depth approach to information security.

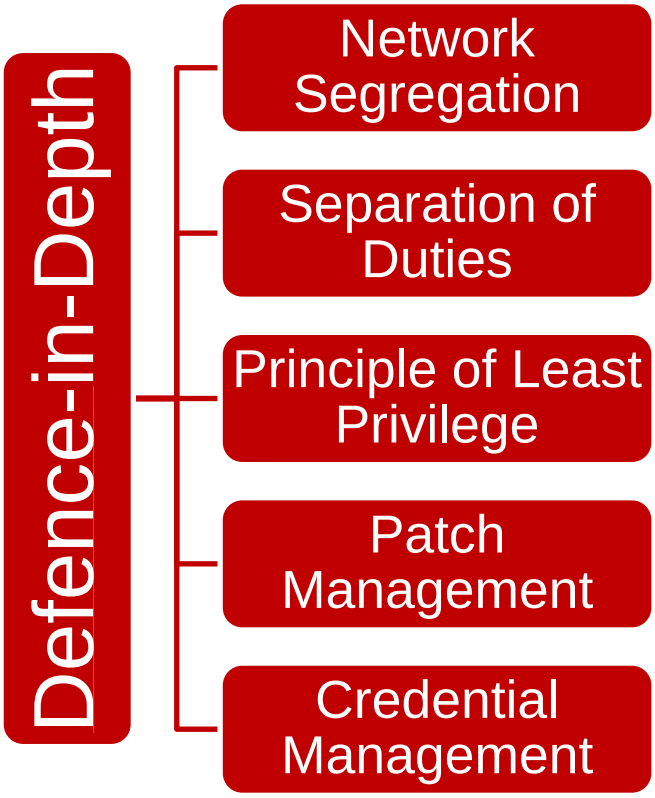

Figure 1: Multi-Phased Security (Defence-in-Depth)

This whitepaper will dissect defence-in-depth into individual components. When all of these components are combined, the likelihood of compromise will be reduced.

Although these are discrete components, they are not mutually exclusive. For example, credential management will be interwoven into all other components, and the principles of least privilege and separation of duties go hand-in-hand. Using multiple components of the defence-in-depth is important when looking to increase assurance.

[6] https://en.wikipedia.org/wiki/Defense_in_depth_(computing)
[7] https://www.fireeye.com/current-threats/what-is-a-zero-day-exploit.html

2.1 Network segregation

Network segregation minimises the network services exposed to various segments of the network, usually on a per-requirement basis.

In addition, it minimises the level of access to sensitive information for those applications, servers, and people who don't need it, while enabling access to those that do. This is usually done by a combination of network devices and network access control.

Below is an incorrectly segregated network architecture commonly encountered by NCC Group:

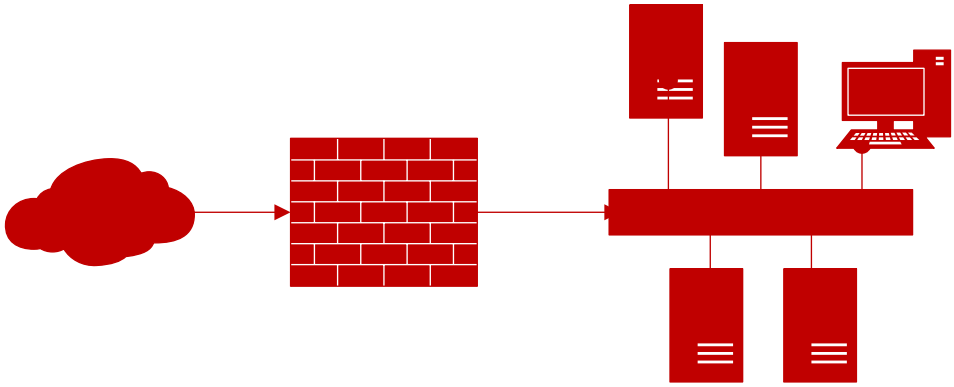

Figure 2: Poor Network Segregation

An attacker compromising the network from the Internet only needs to pass through one external-facing firewall (and when a network is configured "flat", this firewall will often be quite permissive). Once this is done, they're on the internal network. Similarly, should a user's workstation fall victim to client side attacks[8] or phishing,[9] then all servers connected to the same subnet are at an increased risk. All segregation and filtering when done like this comes down to host based firewalls which are usually considerably harder to manage and often isn't done right, if at all.

[8] https://www.honeynet.org/node/157
[9] https://www.microsoft.com/en-us/safety/online-privacy/phishing-symptoms.aspx

Network segregation becomes even more important in a cloud environment:

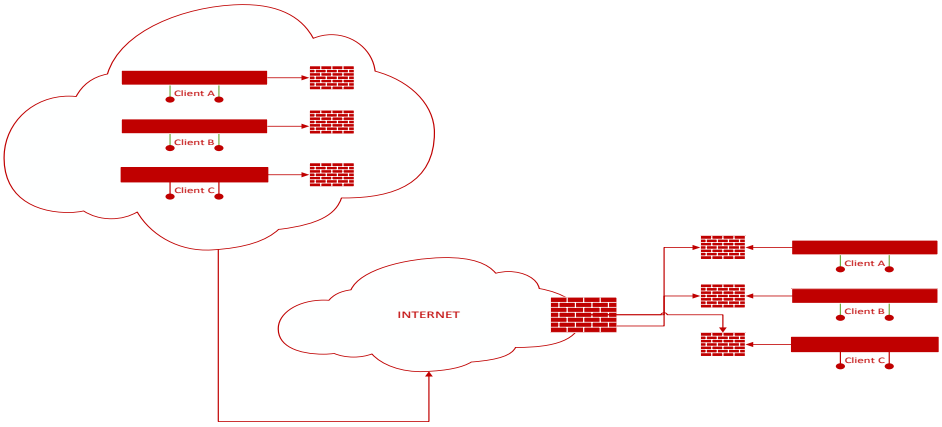

Figure 3: Network Segregation (Cloud)

In this scenario, an appropriate level of network segregation between all tenants is required and while configured, one physical firewall should be configured on devices under the remit of the tenant. Cloud environments are provided with software-based firewalls which can be configured by the tenant. Given the exposure to other tenants and the Internet this is a crucial task when adopting such technologies.

2.1.1 A better solution

Robust network segmentation is achieved through network firewalls, internal routers and host-based firewalls.

2.1.2 Network firewalls

Network firewalls will be used to provide segregation between large, distinct networks such as the perimeter network and the internal network.

Should a malicious user from any mechanism gain remote access to an email server within the DMZ[10], it should not then be possible for them to gain direct access to the internal infrastructure, as network components should be correctly segmented through appropriate network firewalls.

2.1.3 Internal filtering devices

Internal filtering devices can be used to differentiate segments within the same network. For example, if a desktop workstation is compromised, it should prove difficult for a malicious user to pivot to desktops on other segments, as the internal routers should have appropriate filtering enabled[11]. Take the same simple network with enhanced network segregation (namely one firewall, and two internal filtering devices - routers / switches) and things quickly become considerably more difficult for a malicious user attempting to pivot[12] (vertically or horizontally).

[10] http://searchsecurity.techtarget.com/definition/DMZ
[11] There may be legitimate business requirements for different desktop segments to communicate with one another.
[12] https://www.offensive-security.com/metasploit-unleashed/pivoting/

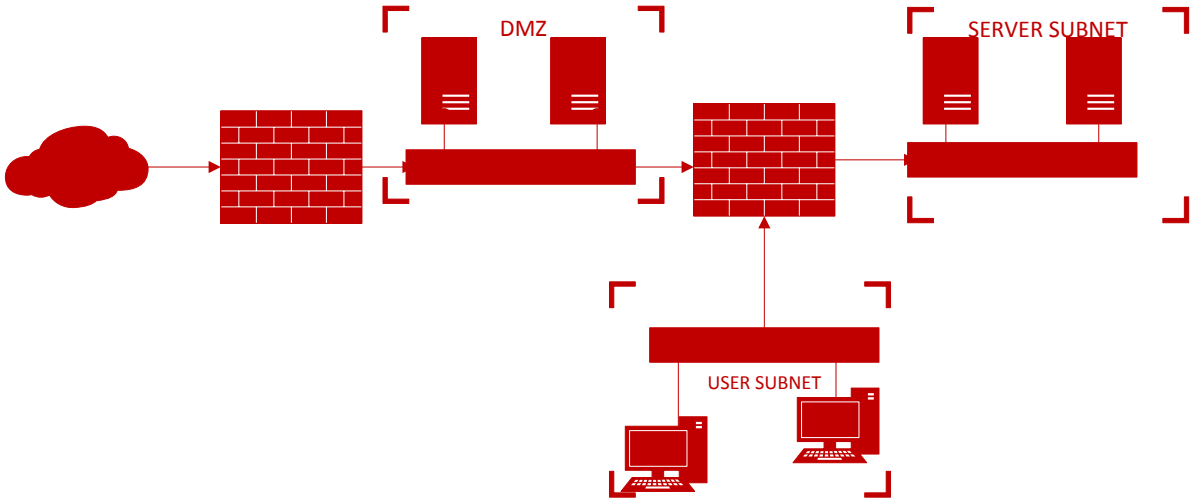

Figure 4: Increased Network Segregation

The Internet-facing firewall can be configured not to process non-HTTP/HTTPS traffic originating from the "User subnet" thus reducing the likelihood of client-side attacks. Similarly, only required accesses can be configured between the DMZ and the server subnet and between the user subnet and the server subnet, as a result reducing the attack surface available should a DMZ machine or workstation be compromised.

Network segmentation is a large undertaking and shouldn't be taken lightly; however, if it can be done correctly, the benefits in terms of security are considerable. It has the added benefit of enabling designers and architects to easily understand data flows across the internal infrastructure to a fine degree.

2.2 Separation of duties

Separation of duties plays a larger role in limiting the impact of a successful attack than preventing an attack in the first place. When combined with identity management, separation of duties can reduce the impact of an attack considerably and prevent further compromise to a network.

A system administrator should usually operate on the belief that vulnerabilities exist within all products even though these may not yet be published (i.e. zero-day vulnerabilities) and this could permit an attacker to compromise that product. Assuming this to be the case, successful implementation of separation of duties could make the difference between a compromise being very limited in impact, or catastrophic.

2.2.1 Identity management

To take this a stage further, identity management is attributing a user's action to a particular person or process. This requires the use of identifiable user accounts within systems to facilitate this attribution.

2.2.2 Separation of duties on a user level

Consider an administrative user logging into a server with domain administrator credentials. Upon logging in, these credentials are often stored in memory, either in clear-text or a notably weak format (a default configuration of Microsoft Windows Server versions prior to 2012). If an attacker manage to gain local administrative access, they could then gain access to certain processes containing such credentials; using these credentials they have now managed to obtain privileges under the context of that user, namely domain administrator.

2.2.2.1 What to actually do?

Consider the same scenario whereby a domain administrator has multiple accounts: a lower privilege account and a highly privileged domain administrator account (e.g. bob and adm_bob). By granting remote desktop permissions or interactive logon permission to "bob" instead of "adm_bob", the clear-text password of "bob" will be the only password leaked. When needing to perform administrative actions "bob" can simply take advantage of the security afforded by Secure Desktop and User Access Control (UAC)[13] thus reducing the likelihood that these administrative credentials are compromised.

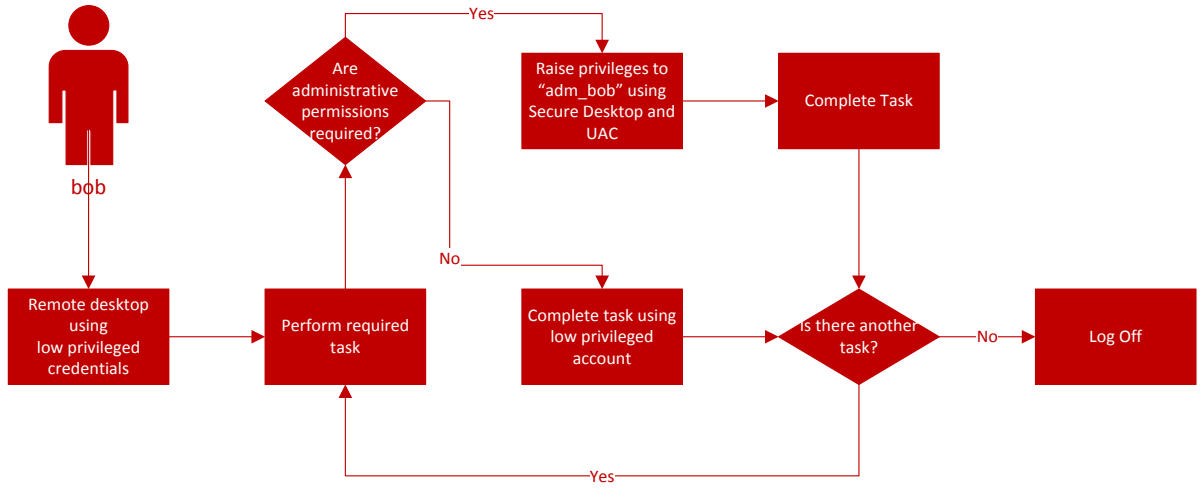

Figure 5: Separation of Duties Data Flow

2.2.3 Separation of duties at a machine level

Similarly, machines should usually be limited to performing one service rather than running multiple services on one server. For example, take a server hosting both Active Directory and Microsoft SQL Server; while the server may be protected by a plethora of network segregation measures, it could still be susceptible to attacks. One attack vector could be a web application vulnerable to SQL injection; an attacker would then have access to the underlying database (often running as SYSTEM – see Principle of Least Privilege and Identity Management). Should the extended stored procedure xp_cmdshell[14] be active (or capable of being re-enabled) then this could allow command execution on the server under the privileges of the SQL process. Running a command such as the one below would have damaging effects.

```
net user attacker @tt@ck3r /domain /add && net group "Domain Admins" attacker /add
```

So an environment that looked reasonably secure and had a large amount of network segregation can be completely undermined by a failure to implement "Separation of Duties".

An attacker will usually take the path of least resistance and when compromising services, they will most likely try to make full use of the server that they compromised before using it to "hop" to another server. If their tasks can be carried out on that server then they will most likely focus their efforts there. Therefore, the separation of duties becomes an important aspect of defence-in-depth.

[13] https://technet.microsoft.com/en-us/library/dd834746(v=ws.11).aspx
[14] http://security.stackexchange.com/questions/2722/xp-cmdshell-should-it-ever-be-used

2.3 Principle of least privilege (Access management)

"The principle of least privilege requires that in a particular environment, every module (such as a process, a user, or a program) must be able to access only the information and resources that are necessary for its legitimate purpose."[15]

2.3.1 Users

Users should be given the least amount of privileges that are required for them to carry out their job. For example, desktop users should not be running as a local administrator. Should a desktop machine be compromised (via a phishing email for example), it would then, in this scenario, be possible for the attacker to extract local hashes and potentially gain access to domain level resources.

Another example, and one that we see all too often, are administrative users using their administrator accounts for non-admin related tasks such as browsing the web. Should a website contain a drive-by-download attack that is successfully able to exploit the client's machine, this would then instantly confer admin privileges to the malicious user. This is further exacerbated by poor network segregation, as the malicious code would have unfettered access to the entire internal network.

2.3.2 Programs and processes

Programs and processes, like users, should also be run with the minimum amount of privileges that are required. If we consider the classic example of a web server, running this process as an administrative service is dangerous and provides little defence-in-depth. If an exploit is found within the web application, then the potential privileges gained are instantly administrative which gives the malicious user further avenues to increase their foothold on the host and then look to abuse other hosts and services.

The principle of least privilege is an important element within the defence-in-depth security stack; when implemented correctly, it immediately reduces the impact should a compromise occur. The primary rule of the principle of least privilege is that users, processes and programs should **always** be run with the least amount of privilege required. If a privilege is required, a change management process should be adhered to such that once the task has been completed the privilege is then removed and a suitable audit event created.

[15] https://en.wikipedia.org/wiki/Principle_of_least_privilege

2.4 Patch management

Patch management involves acquiring, testing, and installing patches to an administered computer system and the software (OS and third party) that resides on that computer.[16]

Patching is primarily broken down into two components: operating system and application or third party patching. A recent report by Praetorian[17] identified the top five methods used by their pen-testers to achieve their objectives, and a lack of patching was not identified as one of these methods. As a result, patching at all levels is considered a cornerstone of the defence-in-depth components and regular, automated patching should always be done.

2.4.1 Operating system (OS) patching

Operating system patching involves patching the underlying software of hosts which includes network devices / firmware, desktops and servers.

Imagine a correctly segmented network with network devices used to segregate distinct areas of an internal network. If these are not kept up-to-date with regards to firmware patching (along with the other principles of least privilege), one exploit could then result in the network segmentation part of the defence-in-depth components being null and void, and the organisation then relying on other factors to provide assurance.

The importance of operating system patching cannot be underestimated. Even when looking at recent patches we can use MS16-032[18] as an example. It is possible to abuse hosts missing this patch[19] and escalate privileges from non-admin users to admin users.

This is especially important for non-obvious traditional desktop and server infrastructure such as embedded systems, even printers, telephony systems or building management systems among many other examples. The key is knowing what is in your computing estate, what it runs and how it can be patched if a security defect become apparent.

2.4.2 Application-layer patching

Third party patching or application patching involves the patching of all software components that are not OS-related. Common examples of third party applications include Adobe Flash, Oracle Java, Mozilla Firefox and Google Chrome web browsers.

Third party application patching has gained significant prominence lately with an increase in phishing and spear phishing attacks[20 &21]. These attacks typically target and exploit weaknesses within third party patching.

A 2015 report by recordedfuture.com[22] discovered that of the top ten vulnerabilities used by exploit kits, Adobe Flash was used in the top eight, and Internet Explorer and Silverlight made up positions nine and ten respectively. This statistic alone shows the value in an organisation ensuring that all third party applications are running the latest secure release.

---

[16] http://searchenterprisedesktop.techtarget.com/definition/patch-management

[17] https://www.praetorian.com/downloads/report/How%20to%20Dramatically%20Improve%20Corporate%20IT%20Security%20Without%20Spending%20Millions%20-%20Praetorian.pdf

[18] https://technet.microsoft.com/en-us/library/security/ms16-032.aspx

[19] https://github.com/FuzzySecurity/PowerShell-Suite/blob/master/Invoke-MS16-032.ps1

[20] http://uk.norton.com/cybercrime-stories-steve/article

[21] http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/solutions/vmware-importance-of-patching-non-microsoft-applications-white-paper-en.pdf

[22] https://www.recordedfuture.com/top-vulnerabilities-2015/

The application of all patches in a timely manner is an important aspect of the defence-in-depth approach to security; as with other principles it requires administrators to fully understand their environment to ensure compliance.

2.5 Credential management

Access to computers, devices and applications is usually achieved through the use of credentials, such as a username and a password or even a second or additional factor of authentication[23]. Many of these credentials are highly-privileged administrative (domain admin / root) accounts. The management of these credentials is paramount in a secure environment.

2.5.1 Password policies[24]

A password policy needs to strike a balance between usability and security. Have no password policy in place and you can be certain that at least one user will have a password of "password". Have an overly complex password policy in place making the resultant password impossible to remember, and you can then guarantee that, once again, at least one user has a post-it note under their keyboard.

By default, a Windows domain password policy requires users to change their password every 42 days; whilst this used to be the recommended value, research carried out by CESG and NCC Group has deduced that placing this burden on end users can have a negative impact on security. By enforcing a longer minimum length and enforcing complexity, this is now considered more secure than an ever changing password[25].

As well as password policies, password filters[26] can be utilised to prevent end users choosing weak passwords. For example, working for "companyA", a password of "companyA123" isn't going to take an attacker long to guess.

2.5.2 Storing passwords

Passwords are usually stored using a variety of methods: encoding, encryption and plaintext. They can be stored in configuration files, or in online or offline password managers.

Firstly, one of the more widely used mechanisms of privilege escalation is plaintext, weakly encoded or poorly encrypted passwords contained within readable configuration files. If a configuration file is required then restrictive permissions should be set on such files to prevent low privileged users from accessing them. Even when passwords are contained within configuration files they should be appropriately encrypted.

Passwords and credentials should always be stored using the best hashing method available. The following trivial example, using Python, shows an **MD5**[27] hash of the word 'password' being calculated. The second example uses **pbkdf2**[28], which is a significantly better hashing algorithm, with the same password. This increases the difficulty and the time taken to extract the clear text password.

```
C:\>py -3
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:24:06) [MSC v.1600 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import hashlib
>>> hashlib.md5(b"password").hexdigest()
' 5f4dcc3b5aa765d61d8327deb882cf99'
>>> dk = hashlib.pbkdf2_hmac('sha256',b'password',b'salt',100000)
```

[23] https://en.wikipedia.org/wiki/Multi-factor_authentication
[24] https://www.cesg.gov.uk/guidance/password-guidance-simplifying-your-approach
[25] https://www.nccgroup.trust/uk/our-research/password-and-brute-force-mitigation-policies/
[26] https://msdn.microsoft.com/en-us/library/windows/desktop/ms721882(v=vs.85).aspx
[27] https://en.wikipedia.org/wiki/MD5
[28] https://en.wikipedia.org/wiki/PBKDF2


```
>>> import binascii
>>> binascii.hexlify(dk)
b'0394a2ede332c9a13eb82e9b24631604c31df978b4e2f0fbd2c549944f9d79a5'
```

When given a choice between online and offline password managers, the choices can often be daunting. With a local password manager, passwords can be stored more securely and on-site rather than having to be share with third parties. Efforts should be made when using a password manager to limit what users have access to which password. Similarly, when accessing a password within a commercial environment it should be appropriately audited.

2.5.3 Default passwords

Applications often come configured with a default password and while this can be easy to remember it will also often be the first guess an attacker will use. All default passwords are published somewhere and are often included within large repositories of default passwords[29].

Publically available tools such as the Metasploit[30] framework allow brute force attacks using default credentials to be executed by attackers with only the most basic knowledge. The following example shows a threaded brute-force tool, created by NCC Group, being used against the "Tomcat Application Manager Login Utility" using default/weak credentials.

```
Dydd_Iau-09-Mis_Ionawr:/home/ed: ./ed.py -h
Usage: ed.py port username password range type

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -p PORT, --port=PORT  port number, default is 8080
  -u USER, --user=USER  username, default is admin
  -P PASSWORD, --password=PASSWORD
                        password, default is admin
  -r NET_RANGE, --range=NET_RANGE
                        class c range to scan
  -t TYPE, --type=TYPE  type of test, options are ssh and tomcat
Dydd_Iau-09-Mis_Ionawr:/home/ed: ./ed.py -p 8080 -u admin -P admin -r 192.168.1.0 -t tomcat
http://192.168.1.3:8080/host-manager/html 400
http://192.168.1.7:8080/host-manager/html 200 admin admin
Dydd_Iau-09-Mis_Ionawr:/home/ed:
```

Figure 6: Example brute force attempt on default password against Apache Tomcat

Taking this into consideration, default passwords should be altered to a unique and complex password. When choosing this password, both the password policy applied and the handling of the password should be considered.

2.5.4 Sharing passwords

Passwords that are shared are often the cause of both horizontal and vertical privilege escalation.

A password should be unique to both its user and its purpose. For example, if an attacker should extract a password for a service account from a configuration file, and this password is unique then the attacker would have to find an alternative attack path; alternatively, if the password is shared with another account, an attacker can then spray the password or hash[31] against a list of known usernames.

[29] http://www.defaultpassword.com/
[30] https://www.metasploit.com/
[31] http://www.commonexploits.com/pass-the-hash/

3 Conclusion

This paper has described the five key principles providing the foundation of a defence-in-depth approach to information security:

- Network segregation
- Separation of duties
- Principle of least privilege
- Patch management
- Credential management

It should be evident that there is no single silver bullet to information security and assurance; a defence-in-depth, multi-tiered approach should be adopted by organisations. It should be ever-evolving and encompass these five key principles to create an environment that has multiple layers of defence that will ultimately provide enough barriers to frustrate malicious users…and maybe the security consultants too.

4 How NCC Group can help

NCC Group, as a global leader in the provision of cyber security professional services and advice, can help organisations in a number of ways, including:

- Security testing/ITHC
- Cyber strategy development and board level education
- Phishing simulation
- Red teaming
- Protective monitoring and outsourced security operations centres
- Cyber incident response and defence operations

To arrange a follow-up, fill in the contact form located at https://www.nccgroup.trust/uk/contact-us/.

5 Further Reading & References

5.1 Further Reading

Cyber Essentials

https://www.cyberstreetwise.com/cyberessentials/

Best Practices for Securing Active Directory

https://technet.microsoft.com/windows-server-docs/identity/ad-ds/plan/security-best-practices/best-practices-for-securing-active-directory

NCC Group Whitepaper on Implementing Password and Brute-Force Mitigation Policies

https://www.nccgroup.trust/uk/our-research/password-and-brute-force-mitigation-policies/

Local network compromise despite good patching: The dangers of NBNS/LLMNR spoofing attacks and how to prevent them

https://www.nccgroup.trust/globalassets/resources/uk/premium-downloads/whitepapers/local-network-compromise-despite-good-patchingpdf/

6 Acknowledgements

The authors would like to thank their colleagues Matt Lewis and Ollie Whitehouse of NCC Group for their peer review and valued suggestions.

