Resources for Chapter 6 – Controls

RESOURCES

- **Evaluating software security patches – Loughborough University, case study**
- **Hacking before and after: How Certified Ethical Hacking (CEH) training changed my perspective on hacking – UCL, case study**
- **Technical vulnerability management**
- **Penetration testing**

Evaluating software security patches – Loughborough University, case study

This case study describes the steps that Loughborough University took to mitigate risk when the Heartbleed OpenSSL vulnerability was disclosed.

Patch Management

Different teams within IT Services at Loughborough University take different approaches to how they manage patching of their services:

- Systems Team who predominantly manage Microsoft Windows Server environments have a cycle where patches are firstly deployed to non-critical systems within a test environment, secondly to critical systems within a test environment. This rollout cycle is then mirrored on production systems.
- Desktop Management Team use SCCM to manage patching on desktop systems. Their approach is to package security patches and updates within SCCM, deploy to a test machine to complete a full Q and A. Once this has been tested, patches will be deployed to IT Services managed machines and then out to the wider University one school at a time.
- Networks Infrastructure Team manages a large estate of Linux servers. As there is no scheduled release of Linux patches, systems use a list to inform administrators of outstanding patches. These are firstly tested within a test environment before being applied to production servers.

When out-of-band patches or patches to prevent zero day attacks are released, IT Security contacts within the department are called upon to evaluate the risk of a potential exploit depending on the systems which are vulnerable and the data stored on these systems.

Depending on the calculated risk, the security team will advise others within the University on how to mitigate this risk. The end goal will be to patch systems, but this is not always possible due to time of release etc.

The security team will also look to leverage border protection systems such as firewalls and IPS/IDS.

Evaluating and patching Heartbleed OpenSSL Vulnerability

On 7 April 2014, Heartbleed OpenSSL vulnerability was unleashed onto the Internet. The security team became aware of this via various sources, which are followed such as Twitter feeds, security blogs etc.

Step 1 – Evaluate and identify

The first step in this process was to try and evaluate the vulnerability and the impact it would have on information systems.

Heartbleed was a vulnerability in OpenSSL, a popular cryptographic library which is used to secure communication across the Internet such as:

- Encrypting HTTPS traffic
- Encrypting emails and Instant messaging services
- VPN encryption

Its most popular use is with Apache web server software, which predominantly runs on Linux server to create HTTPS secure sites. Heartbleed exploits a SSL heartbeat process, in that once a secure connection has been established, periodic pings are sent to keep the encrypted tunnel alive. Specially crafting this ping packet makes the server return 64kb of data from its memory, this could include usernames, passwords, private keys etc. By sending this malicious ping packet multiple times, an attacker is able to rebuild the servers memory contents.

This vulnerability didn't just impact web services; appliances were also vulnerable such as:

- Routers / Switches
- Firewalls
- IPS/IDS
- Load balancers

Due to the nature of this vulnerability, SSL certificates were also deemed as being compromised. This meant that all SSL certificates on vulnerable services would need to be revoked and new certificates issues.

Internally this vulnerability was classified as critical due to the potential level of information disclosure that was possible. Luckily this vulnerability had been disclosed responsibly and patches were already available including a work around. Vendors such as Cisco, Juniper, F5, Palo Alto Networks were also releasing fixes for appliances.

Once the vulnerability was evaluated, the next step was to try and identify vulnerable hosts on the network. Simply getting back the version of OpenSSL would have identified if it software was vulnerable, but we could only do this on servers within our control.

The security team opted to use a script, which had been developed for NMAP, and were very quickly able to identify which hosts on the

network were vulnerable to this exploit.

Step 2 - Communication

From the scan of the network, as expected not all vulnerable hosts and or services were under the control of the department. Other schools and departments around the University were also hosting services.

Internal mailing lists are used to communicate vulnerabilities and releases of patches to the wider IT community within Loughborough University.

As this was more a Linux issue, it was decided to post information to the unix security mailing list. Information about the vulnerability, links to additional information and assistance from IT Services was communicated to IT Support staff across the University.

Step 3 - Patching

At this point, the security team had already configured the IPS/IDS to block any attempts at exploiting the Heartbleed vulnerability externally. This helped remove the external attack vector. The risk remained high due to possible attacks from within the network.

Individual teams patching procedures had already begun within the department, once emergency change requests where approved and testing was complete; the fix was being rolled out onto production services.

Access to management interfaces on appliances followed best practice in that this traffic is completely separate to other business critical traffic and access to this network is heavily restricted to privileged personnel. Due to this, management interfaces were less of an issue and once vendors released updates to resolve the Heartbleed vulnerability, these were scheduled and deployed accordingly.

Step 4 – Follow up

The security team continued to regularly scan the entire network for Heartbleed vulnerabilities. Where services were found to still be vulnerable, service managers were contacted to ensure patches were scheduled to be implemented.

Follow up communications was posted to the internal mailing lists informing server managers that Janet was offering free replacement for SSL certificates. Help was also offered to managers wanting there services scanned to ensure fixes had worked.

Hacking before and after: How Certified Ethical Hacking (CEH) training changed my perspective on hacking – UCL, case study

Austin Chamberlain, ISG UCL

I have been a sysadmin for almost all of my working career. I have a degree in computer science; I worked as a programmer for a year before moving on to system administration, and have worked as a sysadmin for 14 of the last 16 years.

Being a Sysadmin

From a sysadmin perspective, security and hacking are viewed in a defensive or preemptive way. Best practice is to set up servers with strong passwords, regularly patch the servers, and work on the principle of least privilege and privilege separation. These measures are usually passive – once in place, it becomes standard working practice and doesn't require major thought to implement.

In the environments in which I have worked, password security has generally been good. Strong passwords are in use (varyingly) and storage of passwords has been good. Over time storage has improved greatly, with the adoption of tools like Keepass and Lastpass.

Regular patching is usually a work in progress. The structure is simple enough to set up, with WSUS for Windows servers and Satellite for RedHat servers. Even on systems where update processes require more sysadmin intervention, this can be automated with cron scripts or similar.

A frequent problem is older, unmaintained servers – as services are upgraded and servers are replaced, older servers remain in service to host a specific legacy application or code. Sometimes these services are business-critical, but resources are not available to upgrade and migrate the application to a newer system – or the people who wrote and understand the application have left, leaving a key application unsupported and stuck on a legacy server.

Least privilege and privilege separation are key concepts which should be emphasised. In my experience this is generally well handled, since it is mostly under the control of the sysadmin alone. If included as a routine design principle it becomes a habit for sysadmins without being onerous. I have on occasion been surprised where it hasn't been done – where a small amount of effort at the design stage could result in a large increase in overall system security.

Being a hacker is easy

As a sysadmin, I always thought of hacking as slightly esoteric and requiring specialist skills and tools. This is not, in fact, the case. Many of the techniques used are simple and obvious to a sufficiently evil-minded sysadmin. Some of them are relatively common as diagnostic techniques, and I have used some for years. Hacking involves pointing these techniques at systems you don't own. For example, I frequently confirm connectivity between two servers by testing ports with telnet; I'll use banner-grabbing to determine what is listening on a given port. Both of these are techniques used by hackers to get more information about a system to determine which exploits will work.

Hacking tools themselves cover a wide range; at one end there are simple tools which are easily available, or installed by default on many systems. Chaining these simple tools together can be used to compromise a server quite easily. This requires some technical knowledge as many of the tools rely on command-line usage, and usually shell access is the result. The main example of this is netcat, which combined with a vulnerability that allows remote code execution, can be used to establish a remote console connection.

At the other end of the range are the complex tools, which are sadly just as easily available. These don't even require the technical knowledge needed for simpler tools; generally a GUI provides all of the options. Some are perfectly legitimate security scanning/testing tools; nmap and Nessus are good examples, since these will only provide information. Tools like Cain&Abel are slightly more suspect; although there are legitimate uses for this, cracking passwords does not come up that often. Metasploit is the black hat tool of choice; this performs the full spectrum of hacking techniques with a few clicks, ranging from system/service discovery, to compromise, to post-compromise exploitation (shell/desktop access, file transfer, keylogging, screen capture).

Ethical hacking is harder

It is harder to hack in an ethical and systematic way, to produce a useful report on the vulnerability of a given server with the end goal of improving security. Black hat hackers of whatever kind - script kiddies, criminals, foreign government agents - are unlikely to care about the stability of a system unless they are trying to hide their tracks ... and a complete wipe of the system will hide tracks quite effectively.

For this reason ethical hacking combines the diligence of the sysadmin with the creativity and lateral thinking of a hacker. This is not a particularly difficult or rare combination, but it does require a step-by-step procedure of experimental testing, and it takes time to develop both the mindset and the process.

Overall, then, hacking is a growing threat which is becoming more accessible to unskilled users. Protecting against it is strengthened by good procedures, which if they become habit will make any organisation reasonably secure. This can be supplemented and improved by security guidance and penetration testing - but only ever supplemented. A poor base security cannot be improved much after the fact!

Changing Perspectives

- Good practice is vital, and if enforced by policy and inculcated by training, can become habit.
- Patching – automate. Make it routine. Enforce by policy. Do whatever needs to be done to make this happen regularly and frequently.
- Password strength and least privilege design don't require much effort, and greatly increase security.

Hacking is easy and automated. Tools are free and widely available. It's not a matter of if you'll be attacked, it's a matter of how much you're being attacked right now!

Technical vulnerability management

Vulnerability Management

Key Points

- Reduce the risks organisations face resulting from exploitation of technical vulnerabilities.
- Allow organisations to setup a vulnerability scanning framework.
- Assist organisations in developing a patch management policy.
- Support organisations in procuring penetration testing services.

Introduction

A vulnerability is defined in ISO/IEC 27000 as "*A weakness of an asset or a group of assets that can be exploited by one or more threats*".

Vulnerability management is the process in which vulnerabilities in Information Systems are identified and the risks of these vulnerabilities are evaluated. This evaluation leads to correcting the vulnerabilities and removing the risk or a formal risk acceptance by the management of the organisation.

Vulnerability management provides visibility into the risks of assets deployed on the network.

Why we need vulnerability management

A vulnerability management process should be part of an organisation's effort to control information security risks. This process will allow an organisation to obtain a continuous overview of vulnerabilities in their IT environment and the risks associated with them. Only by identifying and mitigating vulnerabilities in the IT environment can an organisation prevent attackers from penetrating their networks and obtaining information

Vulnerability scanners and their risks

As vulnerability management is the process surrounding vulnerability identification, it is important to understand how vulnerability scans are performed and what tools are available. Today, the level of technical expertise required to operate a vulnerability scanning tool is low.

There are risks involved with vulnerability scanning. Since vulnerability scanning typically involves sending a large number of packets to systems, they might sometimes trigger unusual effects such as, for example, disrupting network equipment. In order to cover these risks, it is always important to inform stakeholders within the organisation when vulnerability scanning is taking place.

Roles and Responsibilities

When building a vulnerability management process, the following roles should be considered within the organisation:

- **Security Officer**: The security officer is the owner of the Vulnerability Management process. This person designs the process and ensures it is implemented as designed.
- **IT Security Engineer**: The IT Security Engineer is responsible for configuring the vulnerability scanner and scheduling the various scans.
- **Service Manager**: The Service Manager is responsible for the Information system being vulnerability scanned and the information stored on the system. This role should decide whether identified vulnerabilities are mitigated or their associated risks accepted.
- **IT Systems Engineer**: The IT Systems Engineer role is typically responsible for implementing remediating actions defined as a result of detected vulnerabilities. In most cases, this is likely to be multiple Systems Engineers which could also be spread over multiple teams or departments.

In many organisations the role of the Security Officer and Security Engineer is one.

Vulnerability Management Process

When developing a vulnerability management process, the following phases should be considered:

1. Planning and preparation;
2. Vulnerability Scan;
3. Remediation actions;
4. Rescan.

Planning and Preparation

The first phase in the vulnerability management process is preparation. Initial scans should start with a small scope to prevent being

overwhelmed with hundreds of vulnerabilities. It is important to obtain agreement on which systems/services should be included and excluded from the Vulnerability Management Process.

The first step in this process is defining a scope. The following information should form part of the scope:

- Proposed vulnerability scan date;
- Whether the vulnerability scan is going to be an internal scan or an external scan. An internal scan would be conducted as an authenticated network users, where as an external scan would be conducted from outside of the firewall;
- Whether the scan is going to be an authenticated or unauthenticated. An authenticated scan would be where credentials are provided to login to the application or operating system, whereas an unauthenticated scan would test the authentication process;
- Is this an infrastructure scan or applications scan. An infrastructure scan would check the network footprint of the host or service being tested whereas an application scan would focus on the specified application.

Depending on where the organisation sees the risk will influence the scope; for example some organisations see external threats as the biggest risk and would therefore prioritise Internet facing services.

Once the scope has been defined, this should be distributed to Service Managers. It is very important to get buy-in from service managers and provide them with plenty of notice about upcoming vulnerability scans. It is the responsibility of the Service Manager to liaise with stakeholders to inform them about upcoming vulnerability scans. Depending on the criticality of the system, service managers may have requirement for example not to scan systems during the clearing process.

Plan for unexpected events which might lead to delaying a vulnerability scan depending on the nature of the event. Allow service managers to propose another suitable scan date.

If services are deemed too risky for vulnerabilities scans; this risk needs to be highlighted and the risk accepted by the appropriate senior management. Additional protection will need to be implemented such as ACLs and no external access to mitigate against internal and external threats.

Vulnerability Scan

Once the preparation is complete, the next phase is the initial vulnerability scans are performed. Any issues, which occur during the initial, scans such as systems becoming unavailable or poor application response should be recorded since this may happen on future scans. In this case actions may be defined to reduce the impact of future scans on the stability or performance of target systems.

Vulnerability scanning tools offer a wide range of reporting options to visualize the results. It is necessary to utilize these to create reports depending on the audience:

- **Security Officer/Engineer** Interested in the risk the organisation is currently facing, this risk includes the number of vulnerabilities identified and the severity/risk ratings of the identified vulnerabilities.
- **Asset Owner** Overview of the vulnerabilities in the systems they are responsible for.
- **Systems Engineer** Technical information about the vulnerabilities identified as well as recommendations for mitigation and improvement.

Remediation actions

In this phase, the Service Manager will work with the Security Officer/Engineer to define remediating actions. The Security Officer/Engineer will analyse the reported vulnerabilities and work with Systems Engineers to determine the associated risk and provide input on risk remediation. The risk will depend on factors such as CVSS (Common Vulnerability Scoring System) score for the vulnerability, publicity of the vulnerability, the Security Officer/Engineers personal experience and the classification of information stored on the system.

Vulnerability remediation matrix

Information Classification	Critical Vulnerability	High risk vulnerability	Medium risk vulnerability	Low risk vulnerability
Highly confidential	Remediate	Remediate	Remediate	Remediate
Confidential	Remediate	Remediate	Remediate	Recommended
Not classified	Remediate	Remediate	Recommended	Recommended

Depending on the risk, clear timelines should be provided on when remediating actions should be implemented. Sufficient time should be allowed taking into account the technical nature of the remediation and the organisations change management policies.

If remediation is not possible, this risk should be acknowledged and senior management should made aware. This risk should be documented and accepted via the organisation's risk acceptance process. Compensating controls should be identified in order to mitigate/remove the risk without correcting the vulnerability.

Rescan

Once vulnerabilities have been remediated, a scan should be scheduled to verify the remediating actions have been implemented. The rescan should be carried out using the same vulnerability scanner, configuration and policy. The same reports should be generated as those created during the initial vulnerability scan.

The next set is for the Service Manager and the Security Office to define a schedule on how often a vulnerability scan should be carried out against systems. In order to establish a robust vulnerability management process, it is recommended scheduled scans should be conducted weekly or monthly. This will ensure rapid vulnerability detection allowing the organisation to implement mitigation controls in a timely fashion and reducing the risk.

Patching

Introduction

Patch management is a security best practice designed to proactively prevent the exploitation of known vulnerabilities in information systems within the organisation. The result is to reduce the time and money spent dealing with vulnerabilities and the exploitation of these vulnerabilities. Proactively managing vulnerabilities within information systems will reduce or mitigate the potential for exploitation therefore reducing staff effort in responding after exploitation has occurred.

Patches are additional pieces of code developed to address problems in software. Patches can enable additional features or address security flaws within software. Vulnerabilities are flaws that can be exploited by a malicious entity to gain greater access or privileges. Not all vulnerabilities have patches available; therefore systems administrators must also be aware of other methods of mitigation.

Timely patching of security issues is critical to maintaining the operational availability, confidentiality and integrity of information systems.

Key Principles

New patches are released daily and it is becoming very difficult to keep abreast of all the new patches and ensuring proper deployment in a timely manner. The following high-level key principles can be used to help mitigate the risk of such exploitation.

- The organisation should have a patch management policy, which should be able to identify which patches need to be/have been applied;
- Only software needed to deliver the organisation's business should be installed. This would reduce the number of patches which need to be applied;
- Only the latest stable releases of software should be used;
- The sources of patches should be confirmed and should be evaluated before being applied to the live environment;
- Patches should be deployed as soon as possible to reduce the exposure times to known vulnerabilities;
- A good update and patch process should be encouraged to make it difficult for potential attacks to be successful.

Types of Patches

The organisation should ensure the following information technology infrastructure is covered by the patch management policy:

Type	Patch
Computers/Servers	BIOS, firmware, drivers, hypervisors
Operating Systems	Patches, service packs, feature packs
Application Software (databases)	Patches, service packs, feature packs
Installed Applications (Java, Adobe)	Patches, service packs
Routers and Switches	Firmware
Firewall, IPS/IDS and URL Filtering	Firmware, definition updates
Anti-virus and Anti-spyware	Data files and virus definition updates
Printers and Scanners	Firmware and drivers
Bespoke or in-house developed software	Patches, service packs, feature packs

Penetration testing

Introduction

The purpose of performing a penetration test is to verify the new and existing applications, networks and systems are not vulnerable to security risks which could lead to unauthorised access to sensitive information. A penetration test is also PCI DSS requirement 11.3.

A penetration test should be considered after a vulnerability scan has been completed and any issues identified are resolved or mitigated. A penetration test would identify vulnerabilities, which are unknown or have been missed by the scan. Depending on results it may also highlight where a Vulnerability Management process might be failing.

What is and isn't a penetration test

A penetration test is an authorized, scheduled and systematic process of using known vulnerabilities in an attempt to perform an intrusion into a host, network or application resource. It usually involves the use or automated and manual tools to test resources.

A penetration test is not an uncoordinated attempt to access an unauthorized resource.

Penetration Testing Types

There are two types of penetration test, which can be conducted, black box and white box testing.

- **Black Box** – This form of testing requires no previous information and usually takes the approach of an uninformed attacker. The penetration tester has no previous knowledge about the target system, network or application.
- **White Box** – This form of testing provides information to the penetration tester about aspects of the system or application they are testing. This could be usernames and passwords to access the system, information on how the application is built such as database access etc.

Internal or External Penetration Testing

The threat the organisation is trying to replicate should factor into the decision on how the tests should be conducted. External testing is intended to identify vulnerabilities against hosts and or services, which are accessible via the Internet. Internal testing is intended to identify vulnerabilities with physical access, exposure to social engineering and vulnerabilities to systems, which are accessed via an authorised network connection.

Penetration Testing Scope

Along with the type of testing to conduct, organisations need to decide what they wish to test against, whether it be network testing or a specific application. Other considerations should be whether the testing is conducted internally or externally.

- **Network Penetration Testing** – This will test all services, which are offered by the organisation via the Internet. These include email, DNS, firewall effectiveness and web services. This type of test would also indicate vulnerable software and firewall misconfigurations.
- **Application Penetration Testing** – This type of test could be conducted either internally or externally depending on its availability and testing would be limited to the specified application. An example might be a web application.
- **Social Engineering Penetration Testing** – This type of test focuses on identifying and verifying vulnerabilities associated with employee's ability to understand documented policies and follow procedures and security best practices.

Consideration for using third parties

If the organisation has decided to use a third party to conduct the penetration tests, some effort should be made to confirm the qualification of the company. What are the qualification of the employees and their backgrounds and reference sites. The following penetration testing qualification might be useful to look for when selecting a partner to work with:

- CHECK certified
- Tiger Scheme certified
- CEH (Certified Ethical Hacker) certification

When provisioning a penetration test using an external testing company, ensure a detailed scope of work has been provided and that it meets all of the organisation's testing needs. The following information should form the scope of work:

- What is going to be tested (infrastructure or application test and whether this is an internal or external test) and the type of test (black or white box).
- When is the tests going to be conducted
- Who is the lead contact at the organisation and contact details
- Who is the lead tester and contact details

- Details of any other testers and contact details.

The lead tester and a senior member at the organisation who is authorized to allow the testing to commence should sign the scope of work.

After Penetration Testing

Once the testing is complete, the organisation should request a report, which documents all the vulnerabilities, which were identified by the penetration testing team. This report should also provide remediation advice on how to resolve issues identified.

The report should be held in the strictest of confidence as the report could hold information that would reduce the overall security of the organisation. The organisation should act upon the issues identified as part of a penetration test, this might be implementing remediation steps or accepting the risks and implementing mitigating controls to reduce the identified risk.