In-house Penetration Testing for PCI DSS

GIAC (GPEN) Gold Certification

Author: Jeremy Koster, jeremy.koster@gmail.com
Advisor: Rob VandenBrink

Accepted: 11th of May 2012

Abstract

Many organisations are struggling with the rigorous security requirements that PCI DSS places on those that are storing, processing and transmitting credit card data. One of the tasks that can be difficult to comply with, and costly to outsource, is penetration testing. PCI DSS requires that an organisation perform internal and external penetration testing at least annually and after any significant changes to the environment. This paper attempts to ease the burden of penetration testing by providing methods and sample documents to put PCI DSS compliant penetration testing within reach of the in-house security professional.

1. Introduction

The Payment Card Industry Data Security Standard, introduced in 1999, is a rigorous set of prescriptive requirements aimed at securing systems that handle credit card numbers. The majority of organisations are overwhelmed by the cost of compliance (Ponemon, 2010). Performing security specialist tasks such as formal risk assessment, incident handling, alert monitoring and penetration testing are often over and above the regular duties of the in-house I.T. staff. Maintaining a security team with the capabilities to perform these tasks can be expensive and considered out-of-reach for many small and medium organisations. Smaller organisations with smaller I.T. budgets often need to find ways of lowering the cost of achieving compliance. Penetration Testing, in particular, can be an expensive activity to commission either from an internal team or an external provider. Considering that a clean Penetration Test report is a requirement of PCI DSS (PCI SSC, 2010), an organisation may need to initially perform a number of rounds of testing to achieve a clean report, raising costs again.

Smaller organisations that need to reduce the cost of PCI DSS compliance activities can benefit from taking some of the security specialist compliance activities in-house. The largest hurdle for bringing the Penetration Testing activities in-house is finding a willing candidate within the I.T. team that is prepared to learn the skills and perform the out-of-hours activity that will be required.

This paper aims to provide an understanding of the PCI DSS specific requirements for Penetration Testing, a suitable methodology and sample output documents to allow the motivated internal staff member to tackle the task of penetration testing.

2. PCI DSS Specifics

2.1. Dealing with the QSA

Working harmoniously with the appointed Auditor or QSA (Qualified Security Assessor) is the most important factor in an audit. Clear and open communication is crucial, as a misunderstanding around audit items can lead to expensive remediation work late in the compliance journey. Each auditor has a different approach to reviewing and auditing a penetration test report and may even differ on the interpretation of the standard. This paper tries to find a reasonable balance between a strict interpretation of the standard and the intent of the standard. An auditor commissioned for a particular audit may have different opinions than presented in this paper. It is a good idea to work out your plan for penetration testing well in advance, and to get agreement from the auditor that it is suitable for compliance. A good method of ensuring that compliance activities are on the right path is to provide samples of documentation to the auditor for agreement well before submitting artefacts as evidence. This allows for incremental feedback and provides those being audited with the confidence that their final documents submitted as evidence will pass the audit. It is risky to leave the presentation of reports and intended artefacts to the final stages of audit.

2.2. PCI DSS requirement 11.3

The PCI DSS version 2.0 (PCI SSC, 2010) requires that external and internal penetration be completed at least annually or when there are any significant changes to the environment. The exact wording follows:

> *11.3 Perform external and internal penetration testing at least once a year and after any significant infrastructure or application upgrade or modification (such as an operating system upgrade, a sub- network added to the environment, or a web server added to the environment). These penetration tests must include the following:*
>
> *11.3.1 Network-layer penetration tests*
>
> *11.3.2 Application-layer penetration tests*

This wording of this requirement can be used as a high level description of penetration testing. On the first compliance audit, it is untenable for the auditor to require a penetration test from the year before. It is common practice for an auditor to only require a clean report and evidence of remediation if any exploitable vulnerabilities were discovered. This can be satisfied by running a few rounds of penetration testing the months before audit, which will be discussed later in this paper. It also specifies that penetration testing is required at both the network and application layers. The corresponding Testing Procedures in the standard outline the activities in more detail and gives the penetration tester an understanding of the evidence that will be required. The exact wording follows (PCI SSC, 2010):

> *11.3.a Obtain and examine the results from the most recent penetration test to verify that penetration testing is performed at least annually and after any significant changes to the environment.*
>
> *11.3.b Verify that noted exploitable vulnerabilities were corrected and testing repeated.*
>
> *11.3.c Verify that the test was performed by a qualified internal resource or qualified external third party, and if applicable, organizational independence of the tester exists (not required to be a QSA or ASV).*
>
> *11.3.1 Verify that the penetration test includes network-layer penetration tests. These tests should include components that support network functions as well as operating systems.*
>
> *11.3.2 Verify that the penetration test includes application-layer penetration tests. The tests should include, at a minimum, the vulnerabilities listed in Requirement 6.5.*

The Testing Procedures clarify that the test must be re-run after resolving any identified “exploitable” vulnerabilities. There is some ambiguity in this statement that deserves discussion. Theoretically, all vulnerabilities are exploitable, given the time and resources needed. However, the in-house penetration tester has limited resources and needs to draw the line somewhere with a vulnerability that fails to be exploited. It may be for the reason of technical capability or an intricacy of the vulnerability that is causing the exploit to fail. For this paper, the line is drawn at publicly available exploit code. If the vulnerability cannot be exploited after an hour or two of concerted effort with a corresponding and publicly available exploit code, then the vulnerability is deemed not exploitable. However, all attempts to run the exploit should be documented so that the auditor can get an understanding of the degree of effort and technical skill of the penetration tester. It is generally not feasible for the average in-house penetration tester to write custom exploit code for individual vulnerabilities.

11.3.c requires that the resource must be qualified and possibly independent. The intent of penetration testing for PCI DSS is to protect the environment from malicious parties that may find and exploit vulnerabilities in the target environment to obtain cardholder data. As such, if the penetration tester has experience within the information security field and is familiar with the tools required for penetration testing, then they are a good way to satisfying the qualification requirement. The best qualification for this kind of work is experience in performing a penetration test. Since it is such a dynamic discipline, every penetration test is a learning experience. To gain further experience, without needing all of the red tape, the penetration tester may consider performing a penetration test on an available test or non-production environment.

Organizational independence is also suggested by the standard. If the penetration tester is testing the environment that they are responsible for operating on a day-to-day basis, they may be inclined to make assumptions about the environment or omit certain steps to avoid generating additional workload. It’s important that the penetration tester views the environment from the perspective of an attacker, both external and internal.

11.3.1 and 11.3.2 specify the penetration test layers and indicates particular vulnerabilities that should be tested for in section 6.5. Requirement 6.5 contains general coding vulnerabilities as well as some vulnerabilities specific to web applications. All of these may not be applicable to the target environment but the tools discussed in this paper are capable of testing for them.

2.3. PCI DSS Penetration Test Supplement

Penetration testing can mean a number of different activates, so the PCI SSC has released a penetration testing supplement (PCI SSC 2008) to further clarify what is required of the penetration tester. The points relevant to this paper discussed in the supplement are listed below:

- Who performs penetration testing

- Reporting and documentation
- Scope
- Preparation - Black box or white box testing
- Components
- Important Considerations

Each item is discussed in the below sections.

2.3.1. Who performs penetration testing

The supplement states that the penetration testing can be conducted by a qualified internal resource. To satisfy this condition a penetration tester must have prior and recent experience in performing penetration tests. This can be accomplished by performing multiple rounds of penetration tests on the cardholder data environment ahead of audit time. A prospective penetration tester can perform several penetration tests on the cardholder data environment a few months before it is time to submit the penetration test report to the auditor. A history of penetration testing reports to refer back to, even if it is only over a couple of months, shows that the penetration tester has had the prior experience and has exercised the process of resolving discovered vulnerabilities in the environment. A penetration tester will generally find that the first few penetration tests will result in the finding of Medium and High risk vulnerabilities that require resolution. Resolving these issues can take an organization a significant time to resolve. So planning enough time for the initial penetration tests is important. It is suggested that the first initial penetration test is started at least four months before the planned audit period. This will give the penetration tester enough time to perform two penetration tests, resolve any High and Medium items and produce two reports. By the time that the actual audit arrives, the framework for the penetration test report would have been established and there should be no nasty surprise vulnerabilities popping up that would otherwise derail the compliance efforts. In addition, the final penetration test can utilize the report format and testing methodology from the previous penetration test, saving valuable time and effort close to the crucial audit period.

2.3.2. Reporting and Documentation

The supplement states that the PCI SCC does not have a reporting requirement for penetration test, but does recommend that the penetration test method and results are documented. A penetration test report containing intent, method, findings and has a logical flow through the penetration test steps, is crucial for providing evidence to the appointed QSA. A sample penetration test report containing mock examples is provided in Appendix 5.

2.3.3. Scope

In most cases, the target environment for penetration testing will be a contained cardholder data environment (CDE) with boundaries clearly defined by firewalls. Penetration testing needs to occur on this CDE from an external perspective and an internal perspective. A practical method of determining scope for the external portion is to look at the target environment from the perspective of an outsider with access only to untrusted networks (such as the internet). This will often include a number of Internet accessible systems belonging to the CDE. This will form the basis of the external scope of the penetration test. To determine the scope for the internal portion, the tester can look at the target environment from the perspective of an attacker that has access to trusted networks, but not necessarily from within the CDE itself. For example, the CDE will have a number of users that access the system for administrative and business purposes. The networks used by these personnel to access the CDE can serve as the base for launching the internal portion of the penetration test. The application interfaces and access methods (such as VPN concentrators) can form the basis for the internal scope of the penetration test. Testing the CDE from within its boarders can be beneficial but fails to test the perimeter controls. Additionally, this type of test requires the tester to load exploit tools that may come from unverifiable sources directly onto the internal network of the CDE. Such activities carry a risk and may not be acceptable to the business.

2.3.4. Preparation

This section of the supplement covers how much prior knowledge of the environment the penetration tester should have. This is referred to as black box testing, having no knowledge and white box testing, having an understanding of the environment.

Detailed black box testing will satisfy the PCI DSS requirements but can be a time consuming approach. Gaining some key information about the environment from knowledgeable staff before the penetration test is started will save time. Invariably, the penetration tester will find that they have a better understanding of the environment after the initial rounds of penetration testing. This can lead to the identification of additional targets and result in a more complete penetration test.

The supplement also suggests that the tester can use the material generated as part of other compliance efforts. The most useful of these are; external and internal vulnerability scans, network diagrams and risk assessment results. An approach that comprises of a semi-black box test, later supplemented by internal documentation enables the tester to approach the penetration test with the mindset of an attacker and may reduce the time and effort in the discovery phase.

2.3.5. Components

The supplement encourages the tester to include "social engineering and the exploitation of exposed vulnerabilities, access controls on key systems and files, web-facing applications, custom applications, and wireless connections".

The supplement offers guidance, but does not mandate which components are included in the penetration test. Two that have been omitted without issue in previous penetration tests, have been social engineering and wireless penetration. This paper does not provide guidance on performing these two activities. The selection of these components is highly dependent on the organisation's environment. It's possible that these two areas are heavily utilised and present a significant risk, so controls to protect them should be tested.

2.3.6. Important considerations

This section of the supplement contains some important factors for shaping the nature of the penetration test. Firstly and most importantly, Denial of Service (DoS) attacks are omitted from the requirements of the test. This significantly reduces the scope and impact of the penetration test within the organisation. It also suggests that all relevant parties are made aware of the testing activities. A sample "Rules of Engagement" communication is provided in Appendix 2. The "Rules of Engagement" document explains to those responsible for the environment that DoS testing will not occur. The supplement suggests that the testing is conducted during appropriate maintenance windows and initiated through the normal change control processes. A suggested time frame for penetration testing activities is to perform the less invasive activities between 8pm and 12am, and the more invasive activities between 12am and 4am. This allows the tester to perform most of the penetration testing during reasonable hours, while giving the business the confidence that any system affecting activities will be performed well out of normal service hours. Vulnerability scans are considered less invasive activates. Typically, an environment that is subject to regular vulnerability scans will not be adversely affected by application and network level vulnerability scans. However, it is recommended that first-time vulnerability scans be performed within the 12am and 4am window. Additionally, the support personnel responsible for the operation of the environment should be made aware that there is a higher risk of impact during this exercise. Disruptive or DoS scans should be disabled within the vulnerability scanning tool. More invasive activities include vulnerability exploitation and password guessing. Non-invasive activities such as target discovery, network mapping and vulnerability research can be conducted during normal business hours, as they will not typically subject the environment to abnormal traffic.

3. Tools

Open Source tools have been predominately selected for this penetration test method with the intent of keeping costs down for the in-house penetration tester. While a commercial, point and shoot penetration testing package would probably save time around scanning and reporting, the cost of licensing the software is generally prohibitive. A number of the suggested tools are not as polished as commercial software can be, but they might more closely match what an actual attacker would use to compromise the environment.

When working with the penetration testing tools, it’s a good idea to be familiar with the required commands and the reporting output before using them on production systems. When performing penetration tests on enterprise systems the tester is commonly limited to a testing window of only a few hours in the early hours of the morning. So spending this valuable time sorting out installations, updates or OS issues is not advisable. Be ready to perform the scan, well before the scan, so that the time can be maximised and there is less chance of having to perform the scan again.

A list of tools is located in Appendix 6.

4. Method

The method below is designed to demonstrate a thorough process for penetration testing while still keeping the time spent discovering, researching and exploiting vulnerabilities, to the minimum required. The method explained in this paper borrows heavily from the methods described in the SANS GPEN course (SANS 2012). This course is highly recommended for prospective in-house penetration testers and will assist in convincing the auditor that the penetration tester is sufficiently qualified to perform the testing.

As in the previous section, testing activities are grouped into three categories: Non-invasive, less invasive and more invasive. These categories are organised into three stages respectively.

- Stage 1 – Target Discovery and Network Mapping
- Stage 2 – Vulnerability Scanning
- Stage 3 – Exploitation and Password Guessing

It is important to separate stage two and stage three by a decent amount of time. Performing exploitation directly after vulnerability scans does not give the tester enough time to research and prepare a full range of possible exploits. Research and preparation of exploits can be conducted during normal business hours as it mostly consists of scouring exploit websites and configuring exploit tools in virtual environments. It is recommended that at least a week separates stage two and stage three. This also gives the tester an amount of contingency time to revisit vulnerability scanning if an extra tool is discovered or there was a problem with the results of an initial scan.

4.1. Preliminary Stage - Permission and Notification

Before the first search is conducted or the first tool is executed, the tester must first have permission from the correct individuals and inform the relevant stakeholders how penetration testing will proceed.

4.1.1. Permission memo

A memo stating that you have permission to perform penetration testing is important even for the in-house tester. If there is an issue for any reason, it is good to get a record of the business owner's agreement to the testing activity. Signing a document or requiring that they respond affirmatively to an email is about the best chance of getting them to read and accept the memo. To determine the business owner, it is usually the person in the organisation that the CEO will call in the event of a breach, or if the product is not available to the customer. A suggested permission memo is provided in Appendix 1.

4.1.2. Rules of Engagement

This document is a summary to educate network and system support staff on what to expect in the forthcoming penetration test. It also serves to warn the business how and when you plan to conduct the penetration test. This gives the business an opportunity to alter any of the assumptions that may have made around their preferences on how penetration testing will occur. It also serves to avoid any surprise when notification of penetration testing activities are received during the penetration test. The Rules of Engagement document covers, dates and times, notification of commencement and debrief emails, who is going to be contacted, who to contact in the event of an outage and a summary of the activities that will be conducted.

This communication is also a good opportunity to discuss disabling of any Intrusion Prevention Systems (IPS) if it has been decided to disable IPS for the penetration test. This topic is discussed further in section 4.6.1. Exploitation.

A suggested Rules of Engagement communication is provided in Appendix 2.

4.1.3. Commencement and Debrief emails

The commencement of penetration testing activities will interest all parties involved with the environment and the products that reside within it. It's important to notify network support staff, system support staff, product managers, IT managers and fellow security staff that penetration testing is underway as they may be the recipients of alarms and support calls as a result of the testing activities. They can then call the tester quickly if a problem has been experienced to rule out the penetration test as a source of the problem. It's suggested that the recipients of this notification are broad initially and validated by the product manager and IT manager. If you include a staff member that is not relevant or is not interested in the testing, they will usually tell you after the 3rd or 4th group communication.

Similarly, the same group will be interested in when a penetration testing session has ceased and if there were any preliminary findings. An appropriate way to do this is to distribute a debrief email at the end of the testing session. It is suggested that the debrief email is authored before the penetration test, expanded during the testing and fine-tuned at the end of the session. After a late night of penetration testing, it can be difficult to muster the motivation to author a fresh debrief email.

A suggested format for these communications is provided in Appendix 3.

4.1.4. Contact list

Keeping a contact list of emails and mobile phone numbers allows the tester to quickly communicate with the right personnel in the event of a system issue. This will also help when conducting subsequent tests; the process of identifying the correct personnel will be much quicker. A suggested format for this list is provided in Appendix 4.

4.2. Evidence and Documentation

The most common reason that a tester will have to re-perform a stage of the penetration test is because the evidence of the activity was not captured. Evidence can be in the form of a screenshot, report generated from a particular tool or a standard output on the command line. It's important to be familiar with the keyboard shortcuts for screen grabs and tools used to grab input and output on the command line. It's also important to have a reliable method of copying the evidence to a removable drive for later reference and use in the report. This is especially so if making use of live distributions that don't retain files beyond a reboot. It is recommended that all evidence is time stamped and labelled so that if a finding is contested, a complete record is available to refer to. Ensure that all systems used to perform the test are synchronised to a reliable timeserver.

The best approach is to take too much evidence, use what is needed for the report and archive the rest.

However, the tester should be mindful of the sensitivity of the data that is captured as evidence during the test. If for example, the tester manages to gain access to bulk amounts of credit card numbers, the tester should not retrieve them or even show them on screen. It is recommended that the tester take evidence of filenames and permissions to verify access to files containing bulk sensitive data. If in doubt, check with support staff or the vendor of the target environment to validate that access to the "crown jewels" has been achieved. Retrieving credit card numbers and storing them on the penetration testing machine will introduce implications of PCI DSS scope and places the burden of data protection with the tester.

4.3. Testing resources

To perform penetration from an external perspective and an internal perspective, the tester needs a few resources. Firstly, a laptop that is capable of running LiveCDs and connecting to both wired and wireless connections. If the penetration testing is able to obtain password hashes, then a machine with a powerful processor is also required if password strength is to be tested. More information about tools for this task is provided in section 4.6 that covers exploitation and password guessing.

The other resource that is required, is an Internet service connection not associated with the target environment in any way. It's important to be able to perform the testing activities with the same network conditions as an external attacker would be subject to. Using a network connection that is related to the environment in some way, like through a VPN or a connection directly on an edge router, may provide more or less access than an external attacker would have. This would reduce the validity of the penetration test.

While selecting an Internet service for use with the environment, it's crucial to find an Internet Service Provider (ISP) who is comfortable with the planned testing activities and does not employ blocking in the form of Intrusion Prevention Systems on it's connections. A quick check with the support staff of the chosen ISP may save some frustration during the penetration testing. Some ISPs are stricter than others about traffic that appears malicious on their network.

4.4. Stage 1 – Target Discovery and Network Mapping

This stage of the penetration test involves enumeration of the environment's public facing systems and mapping the network and systems from an external and internal perspective

4.4.1. Web Research

The information that can be gathered from the net can allow the tester to build a profile of how the target environment appears to an external attacker. Services such as Whois (Whois, 2012) and DNS Tools (DNS Tools 2012) allow the tester to identify IP address ranges that need to be tested. Archive.org (Internet Archive, 2012) and Google cache allow the tester to identify components of the network that have since been removed. The classic example is configuration files that were once world readable but have since been locked down.

4.4.2. Network Mapping

This step allows the tester to identify what services are listening on the target IP addresses, preceding network hops and even hosts behind the firewall. Nmap (Nmap, 2012) is the tool of choice for most of these activities. Traceoute is used to discover the network hops and firewalk (Goldsmith and Schiffman, 1998) to perform an analysis of the hosts behind the firewall. A network diagram should be constructed during this step as well as an asset list. These artefacts are included in the final penetration test report and an example of these can be found in Appendix 5. All services identified should be targeted in subsequent steps.

Most target environments are relatively contained, as they have already been through a scope reduction exercise. But if the firewalls and routers involved in protecting the target environment are responsible for large network address spaces there maybe a requirement to ensure that all related IP addresses are scanned for listening services. The process of scanning large sets of IP ranges can be time consuming for the tester. It's important to focus the scanning efforts on the areas that will yield the best possible results. It is recommended that the tester breaks the network targets up into 24 bit subnets and performs fast scanning (switch –F) on the ranges. Fast scanning reduces the scan set to the most well known services. It is rare that an environment will be vulnerable on a port that is not scanned using this method. However, those hosts that have many services listening may require a more thorough scan. It's important that the penetration tester times how long it takes to perform a scan on the environment and quickly calculates how long scanning is going to take. Environments vary greatly on how they respond to a scan. An environment that is configured not to respond to SYN requests, also known as stealth mode, may take a significant time to scan.

4.5. Stage 2 – Vulnerability Scanning

This stage can be split into two parts, network and application level scanning, and web application scanning. The network and application level scanning can be conducted with OpenVAS (OpenVAS, 2012). This tool allows the tester to identify network services that may be vulnerable to attack. Nikto (Nikto, 2012) and OWASP ZAP (OWASP, 2012) can be used for the web application scanning. All these tools generate readable reports that can be sampled in the final report. Each and every vulnerability identified in this stage should be validated by thorough research. Vulnerability scanning suffers from false positives and a number of vulnerabilities may not be applicable to the target systems. This can be for reasons of incorrect versioning, erroneous banners, alternate components and plain mistakes. To validate a vulnerability, first look at it's date. Does it coincide with when the system was last updated? Second, look at the component that the vulnerability resides in, does that exists on the target machine? Third, is this purely based on banner information, is there back-porting occurring that can affect the accuracy of the banner? These questions and more should be asked to ensure that the discovered vulnerability does actually exist on the target environment.

This portion of work, to research vulnerabilities can mostly be conducted during business hours. The tester should plan to allocate a few days between scanning and exploitation to validate all vulnerabilities and ensure that no further vulnerability scanning is needed.

All validated vulnerabilities should be tested in the following stage.

4.6. Stage 3 – Exploitation and Password Guessing

This stage forms the exciting portion of the penetration test. This is where the tester stands the chance of actually gaining a footprint on the environment and beating the protection mechanisms.

4.6.1. Exploitation

Exploitation can mostly be carried out with Metasploit (Rapid7, 2012). For vulnerabilities that don't have a corresponding exploit within Metasploit, a few websites such as exploit-db (Offensive Security, 2012) and inj3ct0r (inj3ct0r Team, 2012) can be used to find exploit code. Exploitation of web applications can be carried out with OWASP ZAP (OWASP, 2012).

It also may be decided to disable Intrusion Protection Systems (IPS) on the target environment as IPS may interfere with exploitation stage of the penetration test. A discussion with the appointed QSA around the expectations of the penetration test will clear up if IPS should be disabled to satisfy audit requirements. The decision may rely on the QSAs understanding of the environment and previous experience with penetration tests. If IPS is disabled the conditions will not replicate what an actual attacker would experience. However, this allows a tester to attempt exploitation of systems that may be vulnerable if the IPS was to succumb to a Denial of Service (DoS) attack.

If IPS is to be disabled, it may be safer only to configure an exception for the IP address where the testing will be conducted from.

4.6.2. Password Guessing

Online password guessing can be conducted with THC Hydra (THC, 2012). It's recommended that the tester acquires a good list of common passwords and potential usernames.

If the tester manages to acquire some password hashes, then John the Ripper (Openwall, 2012) can be used to crack these and Ophcrack (Ophcrack, 2012) if Windows password hashes are obtained.

4.7. Reporting

The tester should allow one to two weeks to write the initial penetration test report. This should give enough time to construct the report, perform a peer review (with information security colleagues), warn the product owners and platform owners of any issues discovered and submit to the appointed Qualified Security Assessor (QSA).

The report should show continuous flow of issue discovery through to resolution. Every item that is identified, as a possible issue, should be followed either to a demonstrable exploit or a non-issue. Issues identified in one stage should be listed at the end of the stage and carried through for further investigation at the next stage. If done in this way, the penetration tester leaves no doubt in the mind of the auditor that a thorough penetration test has been conducted. A sample penetration test report is provided in Appendix 5.

5. Summary

Penetration Texting for PCI DSS is often seen as an onerous and expensive task. While satisfying the PCI DSS requirements is not trivial, Penetration Testing does not always need to be outsourced to an expensive Penetration Testing firm. The methods and examples in this paper form a “how-to” for the technically savvy security professional, with the intent of bringing PCI DSS compliant Penetration Testing within reach.

Appendix 1 – Permission Memo

<Company Name>

Date: <date of issue>

Dear <Product Owner>,

To ensure that the computer systems that are operated by <Company Name> are secure and comply with the Payment Card Industry Data Security Standard, the <Company Name> information security team will be performing penetration testing on the <Company Name> web environment. The penetration testing activities will comprise of network mapping, target identification, vulnerability scanning, password guessing and vulnerability exploitation. While the majority of these activities typically have no impact on the operation of such an environment, the stages involving password guessing and vulnerability exploitation may cause system interruption. No tests specifically designed to interrupt the operation of the computers systems will be conducted. All activities will be conducted in an outage window agreed by the operations and support staff that are responsible for the environment. All relevant and appropriate staff will be informed of the penetration testing activities as they are undertaken and when they have finished.

The purpose of this letter is to inform the <Company Name> management of the penetration testing activities and to gain authorisation for the information security team to perform the penetration testing activities. The penetration testing will begin on the <start date> and finish on the <start date>. A penetration test report will be issued a short time after the penetration test has finished.

Please respond by replying to this email that the Information Security team has authorisation to begin penetration testing. This authorisation is required until <Expected Completion Date>.

Regards,

<Penetration Tester Name>

Information Security

<Company Name>

Appendix 2 - Rules of Engagement Communication

All,

Penetration Testing is scheduled to commence on the <date>. The penetration test will focus on the environment that is in scope for PCI DSS compliance. Penetration testing will be conducted on network elements, servers and applications. Penetration testing will be carried out on the externally facing interfaces of the environment as well as internally facing interfaces. The penetration test will be conducted from the following IP addresses:

External: x.x.x.x

Internal: y.y.y.y

Penetration testing will be conducted in three stages.

1. Target Discovery and Network Mapping
2. Vulnerability Scanning
3. Exploitation and Password Guessing

Stage 1 will be conducted during business hours and does not typically subject the target environment to abnormal traffic.

Stage 2 will be conducted after hours between 8pm and 12am. These activities do not typically cause issues for regularly scanned environments. No scanning activities designed specifically to cause disruption to the environment, such as scans for denial of service vulnerabilities, will be conducted.

Stage 3 will be conducted out of business hours between the hours of 12am and 4am. These activities while not likely to cause disruption to the environment are a more invasive than previous stages.

A change request will be raised to configure the Intrusion Prevention System to allow suspicious traffic from the IP addresses used to conduct the penetration test, during the testing sessions.

Commencement and debrief emails will be sent at the beginning and end of each penetration testing session that is conducted. The list of staff that will be receiving these emails is below:

First Last – name@company.none

First Last – name@company.none

…………

Please respond to this email if corrections or additions of the above list are identified.

This penetration test is not intended to retrieve sensitive information from the environment. Where access to sensitive information is believed to have been gained, the testing will stop and the vulnerability confirmed with the system owners. Any sensitive information inadvertently retrieved by the penetration tester will be securely deleted immediately.

Kind Regards

<Tester Name>

<Tester Contact Details>

Appendix 3 – Commencement and Debrief Emails

Commencement email

Sent to contact list.

All,

Penetration testing will begin tonight at 8pm and finish at 12am. This stage of the penetration test is web application vulnerability testing. The following sites will be scanned for vulnerabilities

<www.target1.none>

<www.target2.none>

<Tester Name> will be conducting the penetration test tonight and can be contact on <phone number> if there are any issues during testing.

A debrief email will be sent out at the end of this testing session.

Regards,

<Sender's Name>

<Sender's Contact Details>

Debrief email

Sent to contact list.

All,

Tonight's penetration test session has been completed. This stage of the penetration test consisted of web application vulnerability testing. The following sites will were scanned for vulnerabilities

<www.target1.none>

<www.target2.none>

A small number of Medium and Low vulnerabilities were discovered by the testing tools. These vulnerabilities will be verified over the coming days and be the subject of further testing.

The next schedule penetration testing session is scheduled for <date>.

Regards,

<Tester's Name>

<Tester's Contact Details>

6. Appendix 4 – Contact List

The below table is an example of a contact list and the responsible positions within the organisation that will need to be made aware of the Penetration Testing activities.

Name	Title	Phone Number	Email address
First Last	Product Manager	01 2345 6789	name@company.none
First Last	Security Manager	01 2345 6789	name@company.none
First Last	IT manager	01 2345 6789	name@company.none
First Last	Network Manager	01 2345 6789	name@company.none
First Last	Platform Owner	01 2345 6789	name@company.none
First Last	Unix Admin	01 2345 6789	name@company.none
First Last	Windows Admin	01 2345 6789	name@company.none
First Last	Network Admin	01 2345 6789	name@company.none
First Last	Firewall Admin	01 2345 6789	name@company.none
First Last	Web Developer	01 2345 6789	name@company.none
First Last	Application Support	01 2345 6789	name@company.none

First Last	Database Support	01 2345 6789	name@company.none
First Last	PCI Project Manager	01 2345 6789	name@company.none

7. Appendix 5 – Sample Report

Penetration Test Report

<Company Name>
<Environment Name>

Introduction

The environment <Environment Names> processes, transmits and stores, credit card information for <Company Names>. As such, <Environment Name> is subject to Payment Card Industry Data Security Standard (PCI DSS) requirements and to maintain PCI DSS compliance. PCI DSS requirement 11.3 requires that Penetration Tests be performed on the Cardholder Data Environment (CDE) that is responsible for handling credit card information. This report documents the method and results of the penetration test conducted on the <Environment Name> environment during the first quarter of <Year>.

PCI DSS Requirement 11.3

PCI DSS 2.0 states the following requirements for Penetration Testing:

PCI DSS Requirements	**Testing Procedures**
11.3 Perform external and internal penetration testing at least once a year and after any significant infrastructure or application upgrade or modification (such as an operating system upgrade, a sub-network added to the environment, or a web server added to the environment). These penetration tests must include the following:	11.3.a Obtain and examine the results from the most recent penetration test to verify that penetration testing is performed at east annually and after any significant changes to the environment. 11.3.b Verify that noted exploitable vulnerabilities were corrected and testing repeated. 11.3.c Verify that the test was performed by a qualified internal resource or qualified external third party, and if applicable, organizational independence of the tester exists (not required to be a QSA or ASV).
11.3.1 Network-layer penetration tests	Verify that the penetration test includes network-layer penetration tests. These tests should include components that support network functions as well as operating systems.
11.3.2 Application-layer penetration tests.	11.3.2 Verify that the penetration test includes application-layer penetration tests. The tests should include, at a minimum, the vulnerabilities listed in Requirement 6.5.

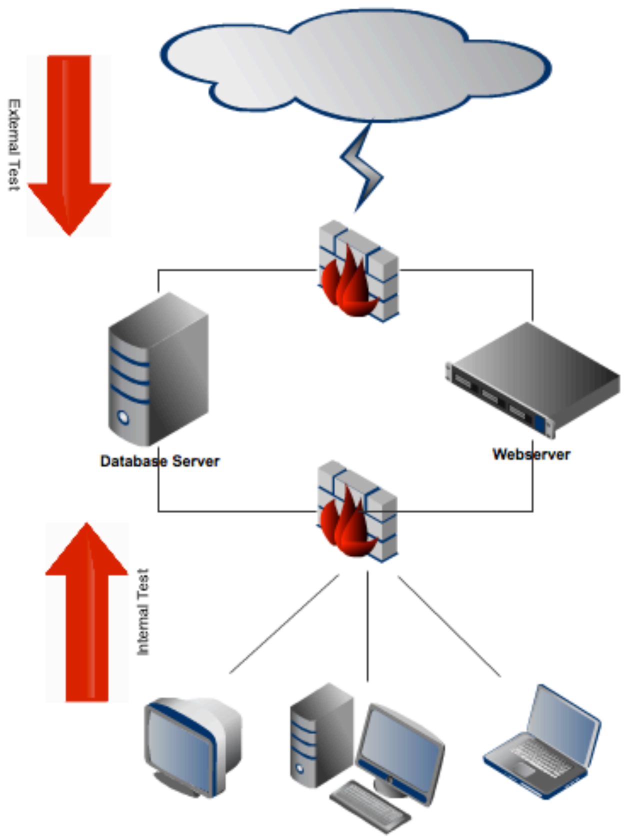
Overview of Method

The penetration test was conducted with the consent of the product owner and all relevant staff members where informed before and during each stage. More invasive tests were performed out of business hours so as to not cause disruption to the daily operation of the environment. The penetration test was conducted on external and internal facing interfaces of the environment.

Target Environment

The <Environment Name> environment processes credit card transactions for <Company> customers and business partners. The environment consists of an Internet accessible website and an application accessible by partner company systems. The environment contained by a perimeter firewall. Support staff access the environment by means of a jump host and VPN.

The following diagram shows an overview of the environment and the external and internal penetration testing approach.



Summary of Results

The penetration test discovered a number of vulnerabilities and ranging in severity. The environment has a high level of risk.

Vulnerability Description	Exploitation	Severity (Risk)
Vulnerability in a webserver application leads to compromise of the server operating system. This may lead to the unauthorised access of all data being handled by this server. This server is responsible for performing over 2000 credit card transactions for product purchase per day.	Remote code execution.	Critical
Vulnerability in the web application leads to	Cross-site	Medium

the ability to perform cross-site scripting. This may lead to the divulging of an individuals credit card number or login credentials.	scripting	

Method

The Penetration test was conducted in three stages:

1. Target Discovery and Network Mapping
2. Vulnerability Scanning
3. Exploitation and Password Guessing

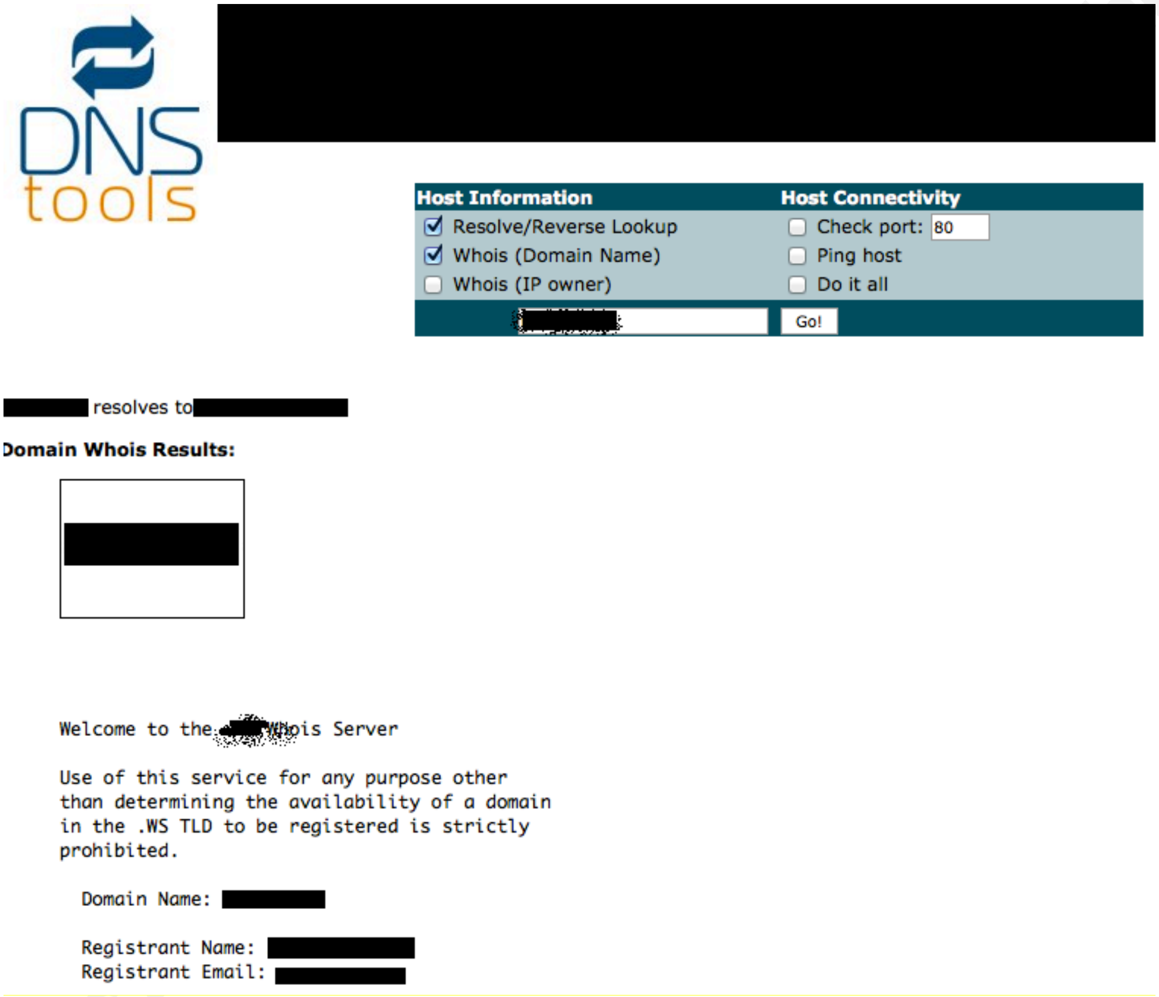
All targets identified in the first stage were scanned for vulnerabilities in the second stage. All vulnerabilities identified in the second stage were validated and exploitation attempted in the third stage. The third stage also included password guessing of the identified user interfaces and network access authentication interfaces, such as VPN concentrators.

Before and after each testing session, informational emails were sent to the following contact list:

Name	Title	Email address
First Last	Product Manager	name@company.none
First Last	Security Manager	name@company.none
First Last	IT manager	name@company.none
First Last	Network Manager	name@company.none
First Last	Platform Owner	name@company.none
First Last	Unix Admin	name@company.none
First Last	Windows Admin	name@company.none
First Last	Network Admin	name@company.none
First Last	Firewall Admin	name@company.none
First Last	Web Developer	name@company.none
First Last	Application Support	name@company.none
First Last	Database Support	name@company.none
First Last	PCI Project Manager	name@company.none

Stage 1 – Target Discovery and Network Mapping

Information that is publicly available about the environment was queried to gain details about the target environment. Details were gathered form DNS Tools, Google Cache and Internet Archive.



[redacted]

Host Information	Host Connectivity
☑ Resolve/Reverse Lookup	□ Check port: 80
☑ Whois (Domain Name)	□ Ping host
□ Whois (IP owner)	□ Do it all

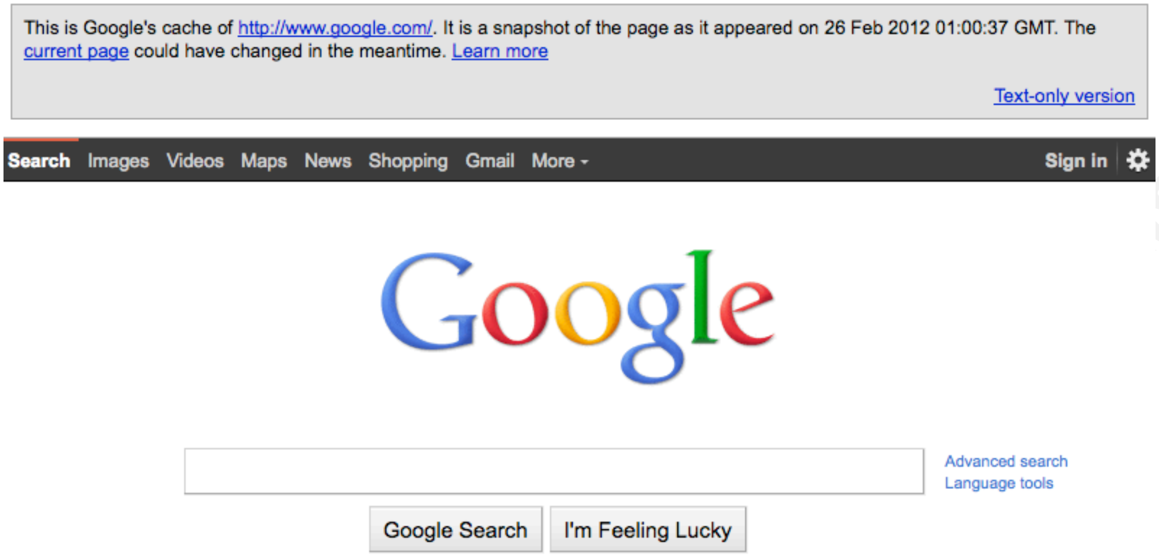
[redacted] Go!

[redacted] resolves to [redacted]

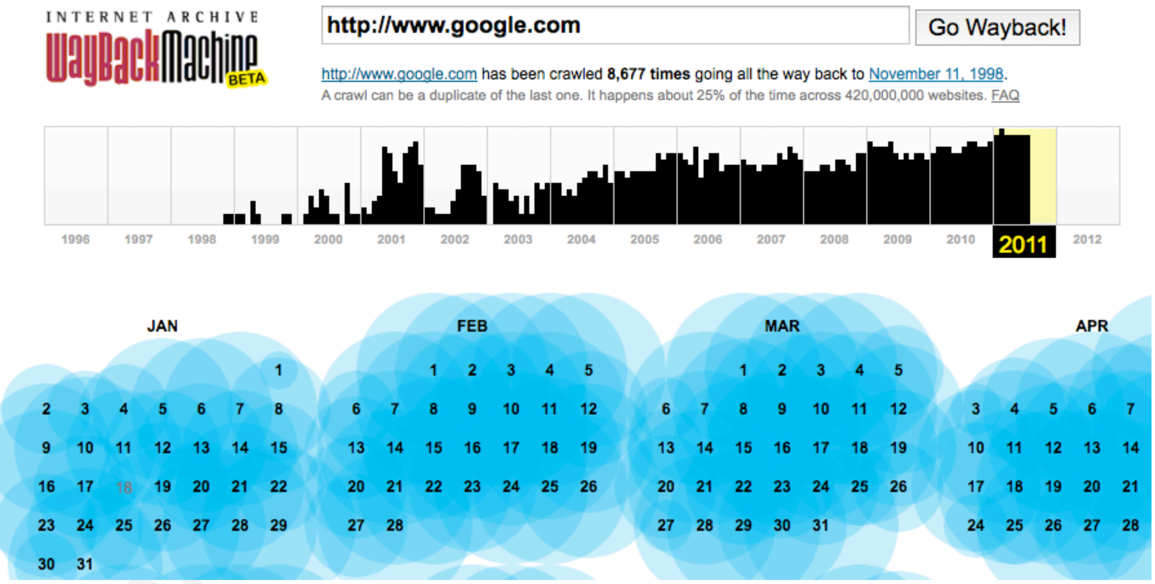
Domain Whois Results:

[redacted]

```
Welcome to the [redacted] Whois Server

Use of this service for any purpose other
than determining the availability of a domain
in the .WS TLD to be registered is strictly
prohibited.

  Domain Name: [redacted]

  Registrant Name: [redacted]
  Registrant Email: [redacted]
```



This is Google's cache of http://www.google.com/. It is a snapshot of the page as it appeared on 26 Feb 2012 01:00:37 GMT. The current page could have changed in the meantime. Learn more

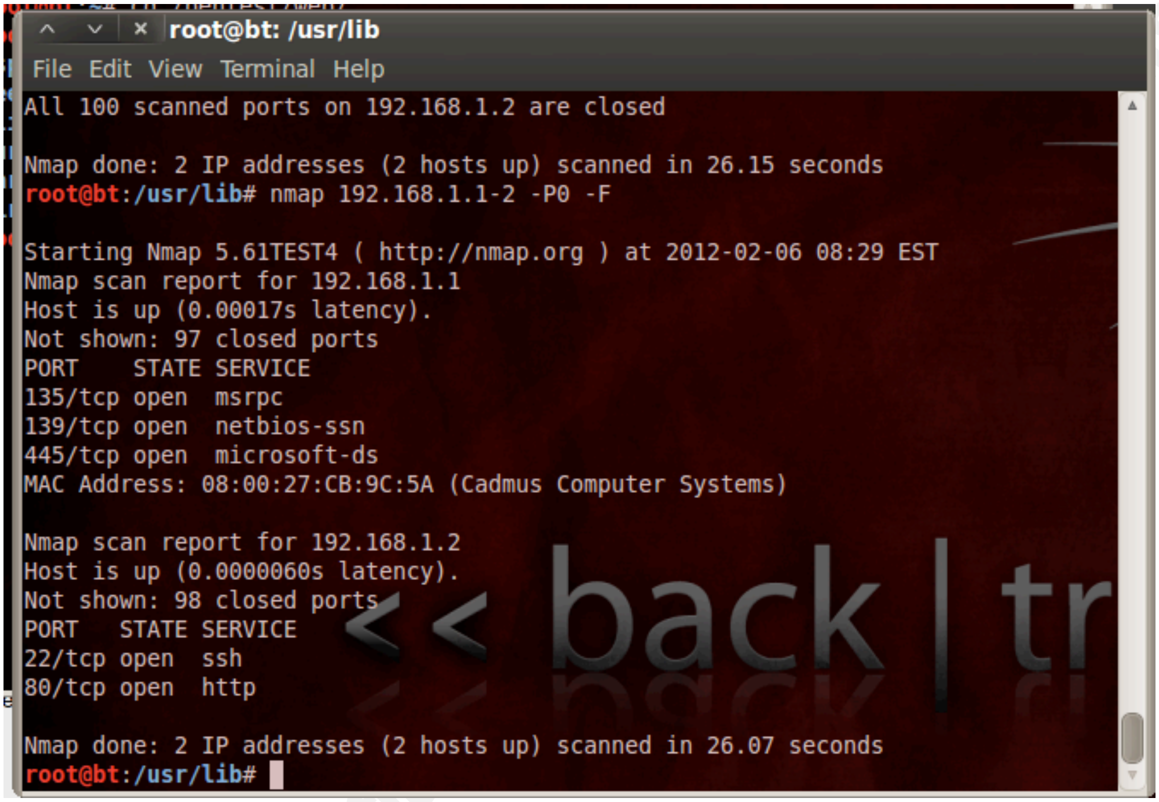


http://www.google.com has been crawled **8,677 times** going all the way back to November 11, 1998.
A crawl can be a duplicate of the last one. It happens about 25% of the time across 420,000,000 websites. FAQ

IP Address Ranges identified:

No.	IP Range	Description	Location	Targets discovered
1	x.x.x.0/24	Web Facing Network DMZ	External	2
2	x.x.z.0/24	B2B DMZ	External	None
3	x.x.y.0/24	Staff access DMZ	External	1
4	x.y.x.0/24	Internal support interface	Internal	1

NMAP was used on identified IP address ranges to gain information about listening services. This was conducted on both the internally facing networks and the externally facing networks.



```
All 100 scanned ports on 192.168.1.2 are closed

Nmap done: 2 IP addresses (2 hosts up) scanned in 26.15 seconds
root@bt:/usr/lib# nmap 192.168.1.1-2 -P0 -F

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-02-06 08:29 EST
Nmap scan report for 192.168.1.1
Host is up (0.00017s latency).
Not shown: 97 closed ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 08:00:27:CB:9C:5A (Cadmus Computer Systems)

Nmap scan report for 192.168.1.2
Host is up (0.0000060s latency).
Not shown: 98 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 2 IP addresses (2 hosts up) scanned in 26.07 seconds
root@bt:/usr/lib#
```

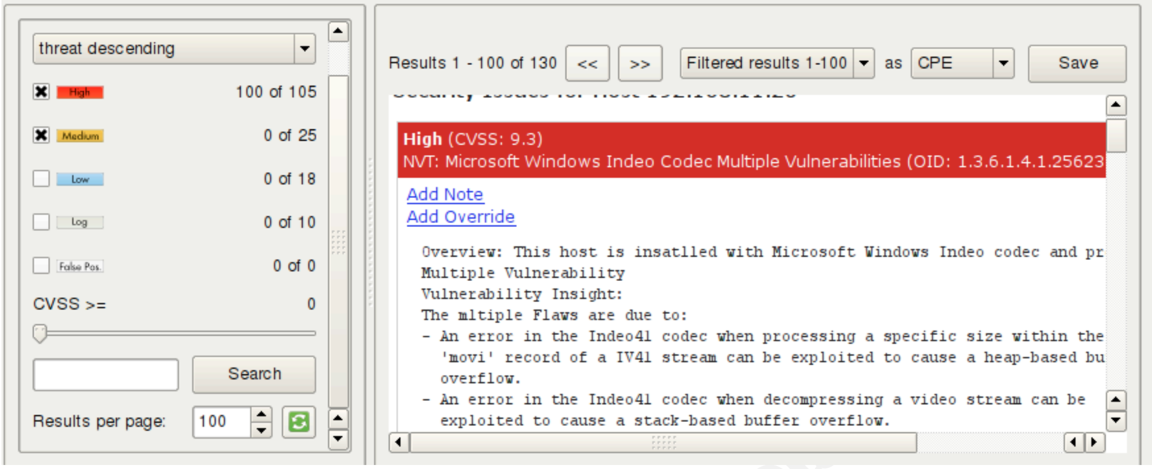
The following targets were identified:

No.	IP Address	Description/FQDN	Location	Services Discovered
1	x.x.x.1	www.target1.none	External	HTTP/HTTPS
2	x.x.x.2	www.target2.none	External	HTTPS
3	x.x.y.1	staff.target1.none	External	IPSEC
4	x.y.x.1	jump.target1.internal	Internal	SSH, RDP

Stage 2 – Vulnerability Scanning

Targets identified in the previous step where subjected to network and application vulnerability scanning with OpenVAS, Nikto and OWASP ZAP.

OpenVAS was used to scan the identified targets. An excerpt of one of the reports generated is provided below.



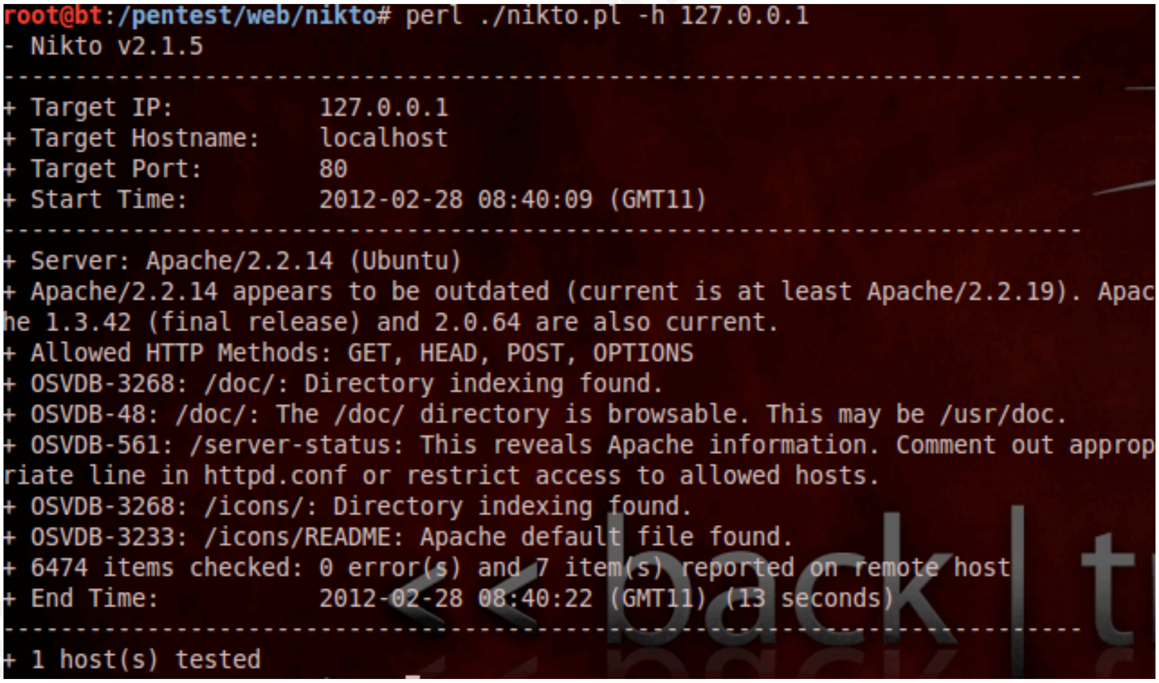
☒ High 100 of 105
☒ Medium 0 of 25
□ Low 0 of 18
□ Log 0 of 10
□ False Pos. 0 of 0
CVSS >= 0
Results per page: 100

Results 1 - 100 of 130 | Filtered results 1-100 as CPE

High (CVSS: 9.3)
NVT: Microsoft Windows Indeo Codec Multiple Vulnerabilities (OID: 1.3.6.1.4.1.25623

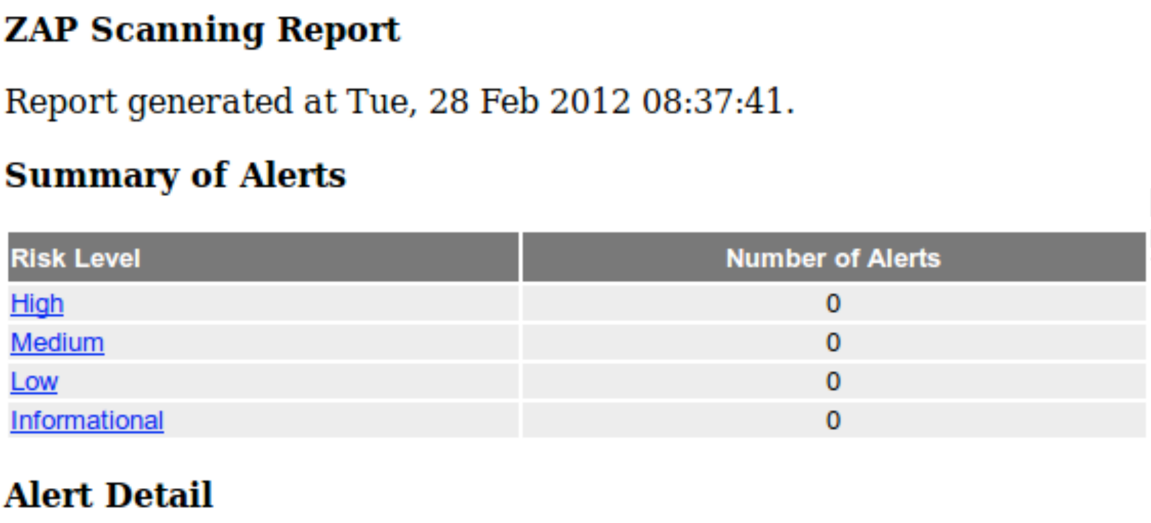
```
Overview: This host is insatlled with Microsoft Windows Indeo codec and pr
Multiple Vulnerability
Vulnerability Insight:
The mltiple Flaws are due to:
- An error in the Indeo41 codec when processing a specific size within the
  'movi' record of a IV41 stream can be exploited to cause a heap-based bu
  overflow.
- An error in the Indeo41 codec when decompressing a video stream can be
  exploited to cause a stack-based buffer overflow.
```

Nikto was used to scan the identified targets. An excerpt of one of the reports generated is provided below.



```
root@bt:/pentest/web/nikto# perl ./nikto.pl -h 127.0.0.1
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          127.0.0.1
+ Target Hostname:    localhost
+ Target Port:        80
+ Start Time:         2012-02-28 08:40:09 (GMT11)
---------------------------------------------------------------------------
+ Server: Apache/2.2.14 (Ubuntu)
+ Apache/2.2.14 appears to be outdated (current is at least Apache/2.2.19). Apac
he 1.3.42 (final release) and 2.0.64 are also current.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-561: /server-status: This reveals Apache information. Comment out approp
riate line in httpd.conf or restrict access to allowed hosts.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 6474 items checked: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2012-02-28 08:40:22 (GMT11) (13 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

OWASP ZAP was used to scan the identified targets. An excerpt of one of the reports generated is provided below.



ZAP Scanning Report

Report generated at Tue, 28 Feb 2012 08:37:41.

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	0
Low	0
Informational	0

Alert Detail

The scanning tools identified the following vulnerabilities.

ID	IP Address	Description	Location	Vulnerability Discovered	Type
1	x.x.x.1	Web application. www.target1.none	External	Cross Site Scripting	Information Disclosure
2	x.x.x.2	www.target2.none	External	Remote code execution	Remote compromise of web server
3	x.x.x.2	Verbose error message	External	The web server gives too much information when handling an error.	Information disclosure
4	x.x.x.4	Windows codec vulnerability.	Internal	Remote code execution.	Remote compromise of server.

Stage 3 – Exploitation and Password Guessing

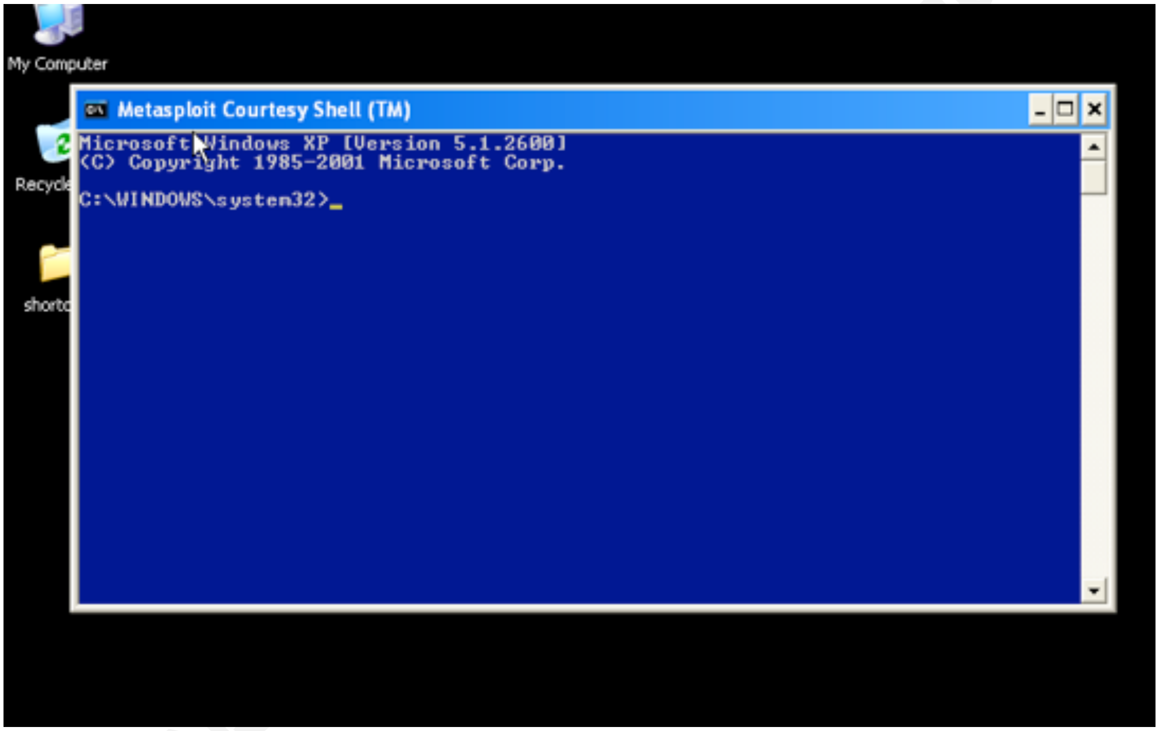
Vulnerabilities identified in the previous stage were researched to validate their existence and identify exploitation techniques. Metasploit was used where there was a corresponding or similar ready-made exploit. Where Metaploit did not have a corresponding exploit, the websites exploit-db and inj3c0r where searched for possible exploit code. Any applicable exploit code was run against the target system.

Vulnerability 1

It was possible to invoke a java script prompt with the use of the identified cross-site scripting vulnerability. This vulnerability was successfully validated.

Vulnerability 2

It was possible to achieve shell on a vulnerable workstation. This vulnerability was successfully validated. A sample screenshot of the exploited vulnerability is shown below.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>_
```

Vulnerability 3

Collaboration with support staff made it clear that the results from vulnerability scanner contained a few false positives. This vulnerability was confirmed as a false positive.

The following vulnerabilities were exploited

ID	IP Address	Description	Location	Type
1	x.x.x.1	Cross Site Scripting. A field in the website could be used to run	External	Information Disclosure

		javascript.		
2	x.x.x.2	Remote code execution. It was possible to execute a utility on the webserver remotely.	External	Remote compromise of web server

Recommendations

The following recommendations are made as a result of vulnerabilities identified during the penetration test.

Patch web server – Immediate Action

The web server application software that houses the web application is of an old version and may be susceptible to known vulnerabilities. This software should be patched to the latest version.

Resolve cross-sight scripting -

Remediate the web page outlined in vulnerability 1 to remove susceptibility to cross-site scripting.

Conclusion

The environment was subject to penetration testing performed by the <Company> Information Security team. Once critical vulnerability was identified and medium vulnerability. The outcome of the results indicate that the environment has a High risk profile. Immediate action should be initiated by <Company> to resolve the vulnerabilities identified in this penetration test report

Appendix 6 – Tools and Resources

Whois

Web-based WHOIS (Whois, 2012) queries can be used to discover ownership details of domains and IP address ranges. When an extra domain is identified within the target environment, details about the associated IP addresses and the domain ownership can be discovered. Most major Internet registries can be queried through easily accessible web applications. Some examples are listed below:

APNIC - http://www.apnic.net/apnic-info/whois_search2

RIPE - https://apps.db.ripe.net/search/query.html

DNS Tools

DNS Tools (DNS Tools, 2012) is a website that allows the tester to perform a number of queries on a target host. The website facilitates, domain name resolution, Whois queries and host / port checking. A number of these services are available to the tester and allow the tester to perform basic lookup functions from a completely independent platform. This can be used to either gather further information or validate target information already obtained.

Archive.org

Archive.org (Internet Archive, 2012) is a website that keeps a backup of many Internet accessible websites on the Internet. Also call the wayback machine, it can be used to find previous versions of a website that may provide clues about previous and existing vulnerabilities. Items such as configuration files, or password files may have previously been removed from the site, but may still contain valid account details.

Google cache

Google cache serves a similar function as Archive.org and may allow the tester to obtain content that was recently removed form the target website or is currently unreachable.

LiveCDs

LiveCDs such as BackTrack (BackTrack, 2012) and Operator (USSysAdmin, 2012) offer a hassle free method of acquiring most of the tools required for penetration testing in an all inclusive distribution. This saves the tester from maintaining their own platform and allows them to use a corporate machine without affecting its corporate build. However, it is not always hassle free. All tools should be updated with the latest versions of definitions, signatures and modules, as the tools included in the LiveCD will almost certainly be out of date. This can take a considerable amount of time and is better to get right in business hours well before Penetration Testing begins. The current version of BackTrack is 5 R1 and can be downloaded from: http://www.backtrack-linux.org/.

Nmap

Nmap (Nmap, 2012) is the open source tool of choice for mapping a network and discovering open ports on ranges of IP addresses. It has an enormous set of functionality but the Penetration Tester need only to use the basic functionality to garner it's core ability of network mapping. The current version is 5.50 and can be downloaded from: http://nmap.org/.

Firewalk

Firewall (Goldsmith and Schiffman, 1998) is a command-line utility written for Unix that uses various TCP/IP protocol techniques to identify live and listening hosts behind a firewall. This allows the tester to gain an understanding of the hosts protected by a firewall. Firewalk is built into most common Linux platforms.

OpenVAS

OpenVAS (OpenVAS, 2012) was originally the open source branch of the once open source Nessus. It is now a decent free option for the penetration tester who has a limited budget. OpenVAS maintains a good set of vulnerabilities that it will scan for at the network and application layer. It will identify most of the common vulnerabilities but may not have the cutting edge vulnerability database that the commercial scanners include. A scan with OpenVAS should be supplemented by web application vulnerability scanners such as Nikto (Nikto2, 2012) and OWASP ZAP (OWASP, 2012). The latest stable version of OpenVAS is OpenVAS 4, and can be downloaded from: http://www.openvas.org/.

Nikto2

Nikto (Nikto2, 2012) is an easy to use, point and shoot perl application that is run from the command line. Nikto checks for vulnerable versions of web servers and associated default files / configurations. It does not look at custom or bespoke applications. It is extensive, but should be accompanied by a scan by a web application scanner such as OWASP ZAP (OWASP, 2012) that looks for XSS, SQL injection etc.

It's installable on any system, which supports Perl, although it can be easier to get running on Linux variants such as Debian or Ubuntu. It is included in the popular live

CDs such as BackTrack (BackTrack, 2012) . The current version is 2.1.4 and can be downloaded from http://www.cirt.net/nikto2
Once downloaded and extracted, ensure that it is updated to the latest plug-ins and databases.

OWASP ZAP

OWASP Zed Attack Proxy Project (ZAP) (OWASP 2012) is a GUI application that is written in java and works like a middleman for your browser, intercepting your browsing traffic and allowing the hacker to manipulate traffic both ways. One of the most useful features is its ability to spider a web application to discover all referenced pages and then scan all the discovered pages for common custom web vulnerabilities. Custom web vulnerabilities are those vulnerabilities that occur in the code of an application as apposed to default files. OWASP ZAP will identify where a web developer has made an error in sanitising input or misconfiguring their web application. It can then export its findings to an easy to read report. The current version of OWASP ZAP is 1.3.4 and can be downloaded from
https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.

Metasploit

Metaspoit (Rapid7 2012) is an exploitation tool and framework that comes bundled with an extensive library of pre-canned exploits. This should be the first stop for the Penetration Tester when identifying an exploit that will match a vulnerability. The most common and reliable vulnerabilities should exploitable by Metaspolit. If the tester has achieved exploitation with Metasploit it is solid evidence that there is problems with the environment. A real-life Metasploit demonstration will convince the most stubborn senior executive that they have to spend money to remediate.

The current fee version of Metasploit is 4.1.0 and can be downloaded from https://community.rapid7.com/community/solutions/metasploit.

THChydra

THCHydra (THC, 2012) is a tool that automates password guessing on user interfaces. It can target a good number of interfaces ranging from website login pages to Windows shares. This tool should be used to test the common usernames and passwords such as vendor defaults that can persist on devices after implementation. The output of this tool will give the tester an understanding of how guessable the usernames and passwords of the system are. The current version is 7.2 and can be downloaded from http://www.thc.org/thc-hydra/.

John the Ripper

John the Ripper (Openwall, 2012) can be used to test the strength of password files that contain encrypted passwords. If the tester is able to obtain a file that contains password hashes, John the Ripper should be run against it on a powerful machine for a few days to ensure that the passwords are not weak. The tester should be aware that this tool, if successful, will expose passwords of a system that may be sensitive in nature. The output of this tool should be treated as sensitive and masked to demonstrate the finding of passwords without revealing the password itself. A good practice is to show the first and last letters of the password but fuzz out the middle letters with an image editor. The files containing clear text passwords should be securely deleted after penetration testing has completed. Passwords that were obtained during the test should be reset by the user to a stronger password. The current version of John the Ripper is 1.7.9 and can be downloaded from http://www.openwall.com/john/.

0phcrack

0phcrack (0phcrack 2012) is a tool that utilises rainbow tables to identify hashed Windows passwords in password files. It is an example of how quickly passwords can be obtained when the tester can get their hands on a windows password file. This can significantly speed up the process of revealing hashed passwords. The current version of 0phcrack is 3.3.1 and can be downloaded from: http://ophcrack.sourceforge.net/.

RainbowCrack

RainbowCrack (RainbowCrack 2012) is another tool that utilised Rainbow tables to crack passwords. There are rainbow tables available for LanMan/NTLM (Windows) password hashes as well as MD5 password hashes. The current version of RainbowCrack is 1.5 and can be downloaded form http://project-rainbowcrack.com/index.htm.

8. References

Back Track Linux (2012, February). BackTrack Linux - Penetration Testing Distribution. Retrieved February 20, 2012 from http://www.backtrack-linux.org/.

DNS Tools (2012, February). DNS Tools | Domain Name Service Diagnosis and Lookup Tools. Retrieved February 27, 2012 from http://dnstools.com/.

Goldsmith, D., and Schiffman, M. (1998, October). Firewalking - A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists. Retrieved February 21, 2012 http://packetfactory.openwall.net/projects/firewalk/firewalk-final.pdf

Inj3ct0r Team (2012, February). 1337day Inj3ct0r Exploit Database : vulnerability : 0day : shellcode by Inj3ct0r Team. Retrieved February 29, 2012 from http://1337day.com/.

Internet Archive (2012, February). Internet Archive: Digital Library of Free Books, Movies, Music & Wayback Machine. Retrieved February 27, 2012 from http://www.archive.org/.

Nmap (2012, February), Nmap - Free Security Scanner For Network Exploration & Security Audits. Retrieved February 20, 2012 from http://nmap.org/.

Nikto2 (2012, February). Nikto2 | CIRT.net. Retrieved February 20, 2012 from http://cirt.net/nikto2.

Offensive Security (2012, February). Exploits Database by Offensive Security. Retrieved February 28, 2012 from www.exploit-db.com.

OWASP (2012, February). OWASP Zed Attack Proxy (ZAP) Retreived February 24, 2012 from https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.

OpenWall (2012, February). John the Ripper password cracker. Retrieved February 20, 2012 from http://www.openwall.com/john/.

OpenVAS (2012, February). OpenVAS - Open Vulnerability Assessment System. Retrieved February 20, 2012 from http://www.openvas.org/.

Ophcrack (2012, February). Windows password cracker based on rainbow tables. Retrieved February 21, 2012 from http://ophcrack.sourceforge.net/.

PCI SSC (2010, October). Payment Card Industry (PCI) Data Security Standard PCI. Retrieved February 20, 2012 from https://www.pcisecuritystandards.org/security_standards/documents.php.

PCI SSC (2008, March). Information Supplement: Penetration Testing (Requirement 11.3 Penetration Testing v1.2). Retrieved February 20, 2012 from https://www.pcisecuritystandards.org/security_standards/documents.php

Ponemon Institure (2010, March). PCI DSS Trends 2010: QSA Insights Report. Retrieved February 24, 2012 from

RainbowCrack (2012, February). Windows and MD5 password cracker based on rainbow tables. Retrieved February 24, 2012 from http://project-rainbowcrack.com/index.htm.

Rapid7 (2012, February). Metasploit Penetration Testing Software. Retrieved February 20, 2012 from http://www.metasploit.com/.

SANS (2012, February). SANS GPEN (GIAC Penetration Tester) course. Retrieved February 20, 2012 from http://www.giac.org/certification/penetration-tester-gpen.

THC (2012, February). THC-HYDRA - fast and flexible network login hacker. Retrieved February 21, 2012 from http://www.thc.org/thc-hydra/.

USSysAdmin (2012, February). US System Administration » Operator. Retrieved February 27, 2012 from http://www.ussysadmin.com/operator/.

Whois (2012, February). Whois - Wikipedia, the free encyclopedia. Retrieved February 27, 2012 from http://en.wikipedia.org/wiki/Whois.