Web Application & Penetration Test Report

Prepared for:
****

Scope

Target(s)

The scope of the test included the following in-scope information assets:

10.0.0.0/24
10.0.1.0/24
10.0.2.0/24
192.168.1.0/24

Control(s)

The in-scope information assets were measured against the following controls:

- Open Web Application Security Project (OWASP)
- Penetration Testing Execution Standard (PTES)

Timetable

The following testing timetable is shown below:

- Test Start: ****
- Test End: ****

Scope

Overview

MalwareCare has adopted an industry-standard approach toward security assessments. This approach is used in all our assessments and provides our clients with real-world risks that take into account a number of factors ranging from: Skill Level, Motive, Ease of Exploit to Financial and Reputational Damage. Our comprehensive approach ensures that our clients' vulnerabilities are represented by their true real-world likelihood and potential impact to their business.

MalwareCare conducted a Network Penetration Test against the organization using a methodical and standardized approach. The objective of the assessment was to measure the security posture of the in-scope assets and identify any deviating vulnerabilities by measuring them against industry-adopted controls. For more information about our approach and methodology, please see Appendix A.

Important findings from the assessment were communicated to management either during or following the assessment as appropriate based on the nature and risk level of the finding. All of our findings are explained in detail in the Findings section of this report.

Summary

We were engaged by the client to perform an internal network penetration test. As a result of this test four (4) vulnerabilities were identified. One (1) of the findings was identified to be critical risk. Another one (1) was found to be high risk.

The critical finding is attributed to each of the systems sharing the same administrator password. By gaining access to one system on the network we were able to leverage the administrator password on that system to log into the rest of the computers on that network. Due to the impact of an attacker gaining access to all ofthe computers on the network this finding was rated as critical. To mitigate this attack each computer should have a local administrator account that has a unique password.

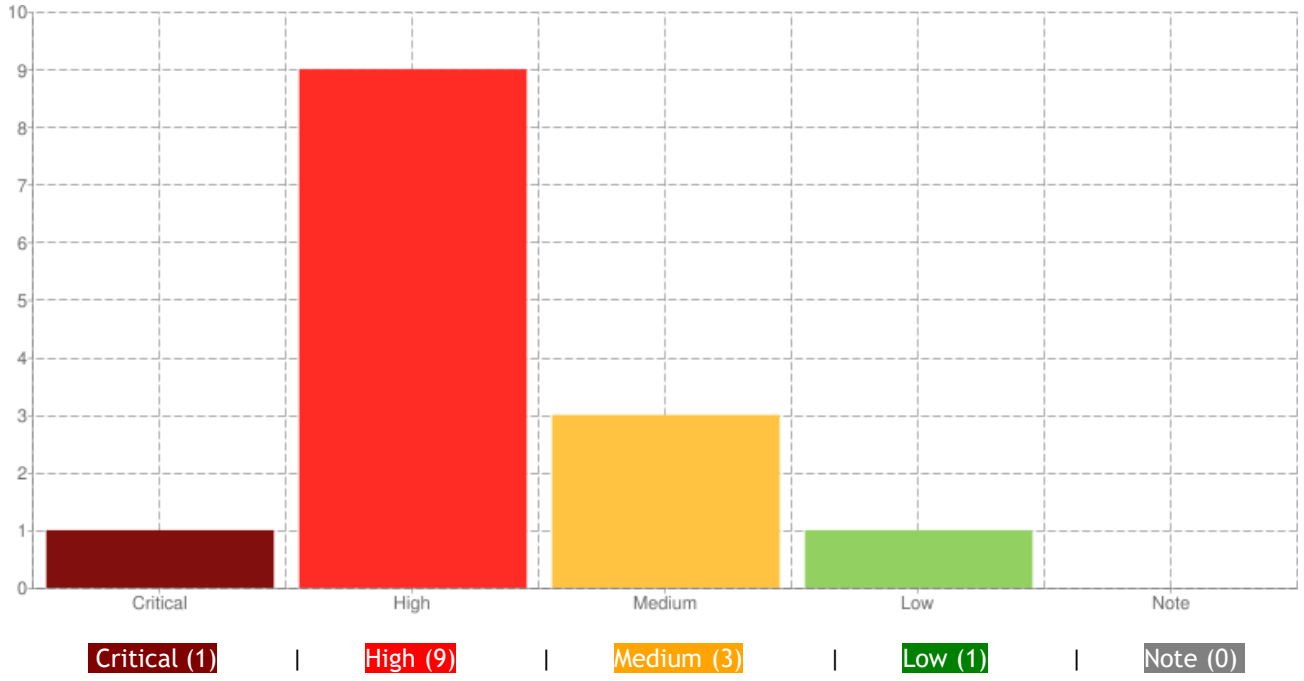
The high finding is attributed to a misconfiguration of the service Server Message Block (SMB). The service is currently configured to not sign messages. The signing ensures that a computer responding to SMB broadcast messages is who they say they are. Because signing is not enabled, it is possible for any other computer on the network to respond to SMB broadcast messages. By doing this an attacker is able to add malicious content to someone else's message, or respond and ask for credentials. Due to the fact that automated tools have been created for this attack, but it must occur from within the network, this finding has been rated as high. Each of the affected systems should be configured to enforce SMB signing.

Each of the other findings, medium and low risk, are associated to common misconfigurations. These findings have been risk ranked this way due to relatively low impact or likelihood of exploitation. It is our recommendation that each of these findings be reviewed and mitigated as appropriate, as these findings represent risk to the environment.

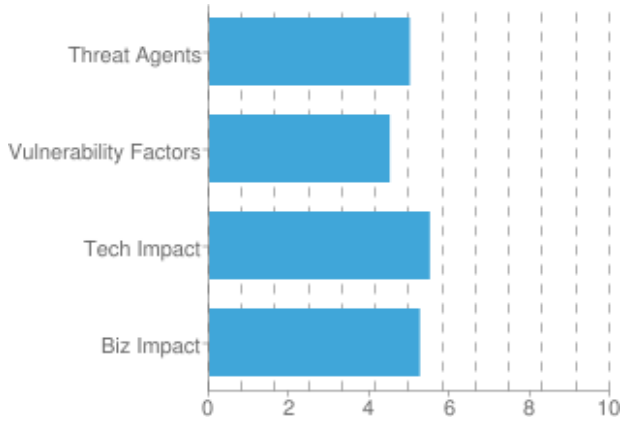
Scope

The charts below are designed to provide a quick snapshot of the assessment. For information regarding risk ratings, please see Appendix B. Otherwise, for vulnerabilities as a result of this assessment, please see the Findings section.

Total Vulnerabilities by Rating



Average by Risk Factor



Average Overall Rating

MEDIUM

Scope

Quick View

The table below is designed to provide a quick view of all the identified findings and their respective risk ratings. Please see the following section for a detailed listing of the identified findings.

For information regarding our risk rating methodology, please see Appendix B.

#	Finding Title	Instances	Rating
1.	Shared Local Administrator Password	1	Critical (8)
2.	SMB Signing Not Enabled	9	High (7)
3.	DNS Cache Snooping	3	Medium (4)
4.	Apache mod_negotiation (Apache MultiViews)	1	Low (1.25)

Total Findings: 14

Scope

1. Shared Local Administrator Password | Critical (8)

Description:

During the internal testing, it was determined that the local Administrator password is shared among more than one computer. The local Administrator account is installed by default on Windows with the password set during the operating system setup. The account has full access to all files on the system.

Impact:

Generally, automated tools are used to install Windows in larger organizations. This causes an issue since all of the local Administrator passwords are the same unless changed after installation. If an attacker were to gain access to one system and gain the local Administrator password or hashed password (encrypted password) then all systems could easily be compromised. This is one of the most prevalent avenues for an attacker to pivot and escalate inside of an internal network.

Test(s) Conducted:

After accessing a system, each username and password hash is used from the system to attempt to log into other systems.

Finding Comments:

MalwareCare was able to use a Metasploit module called PSEXEC to perform a pass-the-hash attack against each of the systems within the 192.168.1.0/24 network. This module allowed for testers to remotely gain access to the systems because of a shared administrator password.

Recommendations:

Utilize a solution that changes all local Administrator passwords regularly. LAPS (local Administrator password solution) is a tool which could be used to remediate this vulnerability. Alternatively, some other enterprise level password management tools can also help ensure you are not using shared passwords.

Affected System(s):

192.168.1.0/24

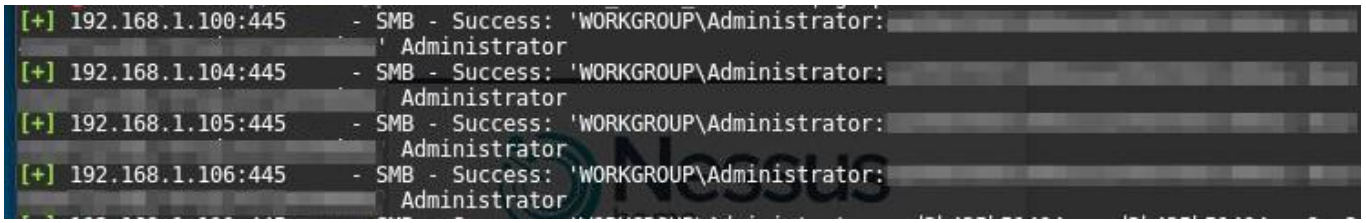
Instance(s):

1

Status:

Not Remediated

Evidence:



Evidence notes:
The above screen capture shows that MalwareCare was able to successfully authenticate to multiple systems usingthe same username and password.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

Critical (8) = (Likelihood (8 + 8) /2 = Critical (8) + Impact (8 + 8) /2 = Critical (8)) /2

Reference(s):

https://www.microsoft.com/en-us/download/details.aspx?id=46899
https://www.offensive-security.com/metasploit-unleashed/psexec-pass-hash/

CVSS:

(AV:N/AC:L/Au:S/C:C/I:C/A:C)

[Back to Top]

Finding(s)

2. SMB Signing Not Enabled | High (7)

Description:

It was identified that the local network utilizes the Server Message Block (SMB) and NetBIOS Name Service (NBNS) protocols without signing. The SMB protocol is used to provide network file sharing and communication between nodes on the network. NBNS is similar to the Domain Name System (DNS) protocol in that it translates human-readable computer names into IP addresses. Both of these protocols are used in authentication between a domain-connected computer and a Domain Controller. By not having signing for these messages enabled it is possible for an attacker to alter legitimate messages, or send their own.

Impact:

Successful exploitation can enable the attacker to capture usernames as well as hashed passwords, piggyback on the initial SMB request to include a command or payload, and direct a victim to the IP of an attackers choosing.

Test(s) Conducted:

A system is setup in promiscuous mode to listen for NBNS requests. When a request is captured, a race condition ensues and a response is spoofed to the originating system before the Domain Controller replies. As a result, the victim system will attempt to authenticate by sending it’s SMB credentials to attacker’s system.

Finding Comments:

While reviewing the configuration for SMB for each of the internal systems, it was identified that several of them did not require SMB signing. By listening for this traffic and setting up our system to respond to broadcast SMB traffic it was possible to capture usernames and password hashes.

Recommendations:

Ensure that passwords are sufficiently strong, in order to mitigate an attacker’s ability to crack the hashes that are captured. SMB signing should also be enabled, which prevents an attacker from spoofing the response to a system. SMB signing does have some impact on use with legacy systems, and should be reviewed before enabling.

Affected System(s):

10.0.0.100
10.0.0.101
10.0.0.102
10.0.0.136
10.0.1.101
10.0.1.102
10.0.2.100
10.0.2.101

Finding(s)

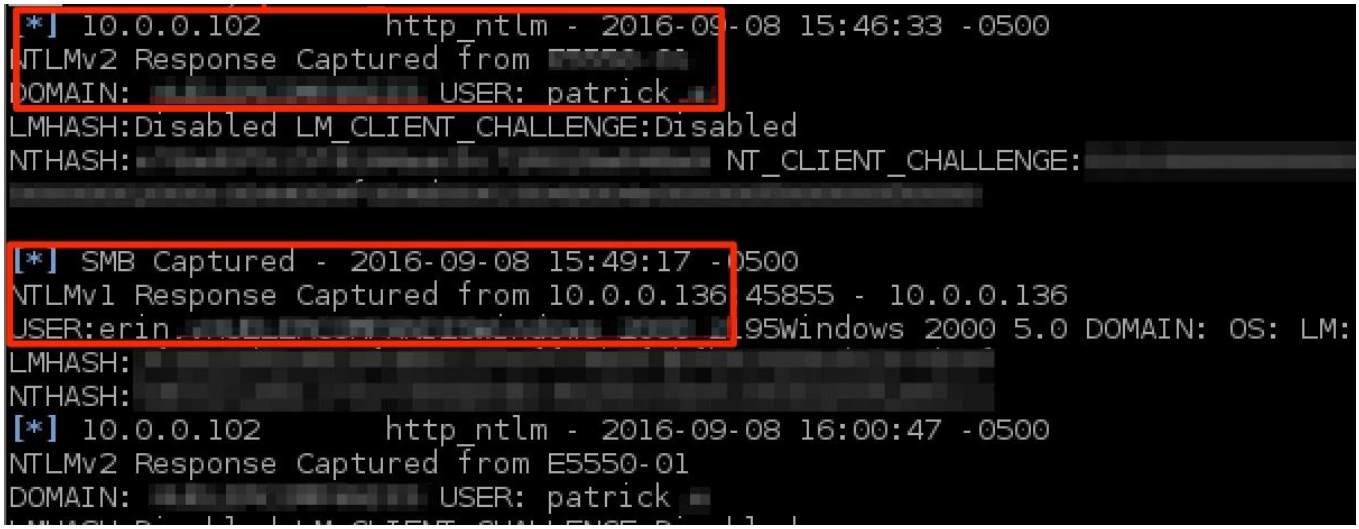
Instance(s):

9

Status:

Not Remediated

Evidence:



Evidence notes:
The above screen capture shows user systems responding to a man-in-the-middle attack, because SMB signing is not enabled.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

High (7) = (Likelihood (6 + 8) /2 = High (7) + Impact (7 + 7) /2 = High (7)) /2

Reference(s):

http://www.packetstan.com/2011/03/nbns-spoofing-on-your-way-to-world.html
https://www.fishnetsecurity.com/6labs/blog/path-least-resistance

CVSS:

(AV:N/AC:L/Au:N/C:C/I:C/A:N)

[Back to Top]

Finding(s)

3. DNS Cache Snooping | Medium (4)

Description:

Domain Name Systems (DNS) are used to resolve a server's name to an IP address. These systems often keep a record of what names and IP addresses have been resolved to make those look ups take less time in the future. By making several requests to the DNS server, and setting the Recursion Desired (RD) to zero, it is possible to enumerate a list of systems and/or websites that have been cached by the server.

Impact:

An attacker is able to use this list of systems and websites to target craft attacks based on the URLs found. For example, references to antivirus updates are a good indication the antivirus is in use. References to social media, or blogs may be spoofed to capture sensitive information.

Test(s) Conducted:

A connection is made to the DNS server, and several requests for common URLs (i.e. www.youtube.com, www.facebook.com, www.linkedin.com) are performed. Any valid response from the server indicates it has that resource cached. Otherwise the request is forwarded to another DNS system that may know where the resource is located.

Finding Comments:

Each of the servers are configured in such a way to allow for an attacker to query the DNS service for cached resource records. By obtaining this information an attacker can begin to build a profile for sites they may want to clone for use in additional attacks, such as social engineering.

Recommendations:

Disable recursion within the DNS server's configuration. If the configuration cannot be directly changed, contact the vendor for any possible updates or work arounds.

Affected System(s):

10.0.0.2
10.0.1.2
10.0.2.2

Instance(s):

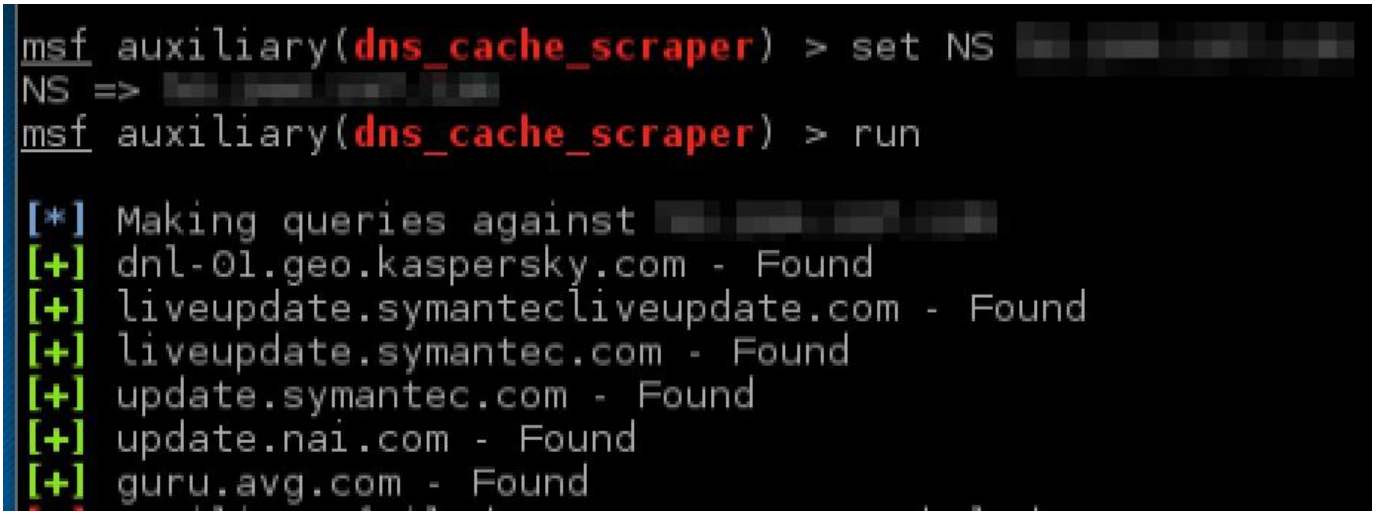
3

Status:

Not Remediated

Evidence:

Finding(s)



Evidence notes:
The above screen capture shows results for cached domain names in the server. The information above indicates a likely use of Symantec as the corporate anti-virus.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

Medium (4) = (Likelihood (5 + 1) /2 = Low (3) + Impact (5 + 5) /2 = Medium (5)) /2

Reference(s):

https://www.acunetix.com/vulnerabilities/web/dns-cache-snooping
https://support.microsoft.com/en-us/kb/2678371

CVSS:

(AV:N/AC:L/Au:N/C:P/I:N/A:N)

[Back to Top]

Finding(s)

4. Apache mod_negotiation (Apache MultiViews) | Low (1.25)

Description:

mod_negotiation is an Apache module responsible for selecting the document that best matches the client's request from one of several available documents. If the client provides an invalid Accept header, the server will respond with a 406 Not Acceptable error containing a pseudo directory listing.

MultiViews is an Apache option which acts within the following rules:
If the server receives a request for /some/dir/example, if /some/dir has MultiViews enabled, and /some/dir/example does not exist, then the server reads the directory looking for files named example.*. Next, the server creates a type map which lists all those files with the same name, assigning them the same media types and content-encodings it would have if the client had asked for one of them by name. After the 406 error, the server returns the best match(es) based on the request.

Impact:

mod_negotiation can help an attacker to learn more about the target and, for example, generate a list of base names, generate a list of interesting extensions, and look for backup files.

Test(s) Conducted:

Perform manual HTTP requests and modify one of three accept headers (Accept, Accept-Language, and Accept-Encodings). By modifying one of these headers to an invalid MIME type and a filename prefix in the URI the server will respond with all files within the MultiViews configured directories in a HTTP response Error 406.

Finding Comments:

The Apache server is currently configured to allow multiViews. By sending a request to the server for a specific file that does not exist, the server responds with a directory listing of files with a similar name. This can allow for an attacker to begin querying the web server for any files that are hosted, but not accessible by the web site. Examples that we commonly look for include backup files, config files, and readme.txt.

Recommendations:

Disable the MultiViews directive from Apache's configuration file and restart Apache.

Affected System(s):

10.0.1.123

Instance(s):

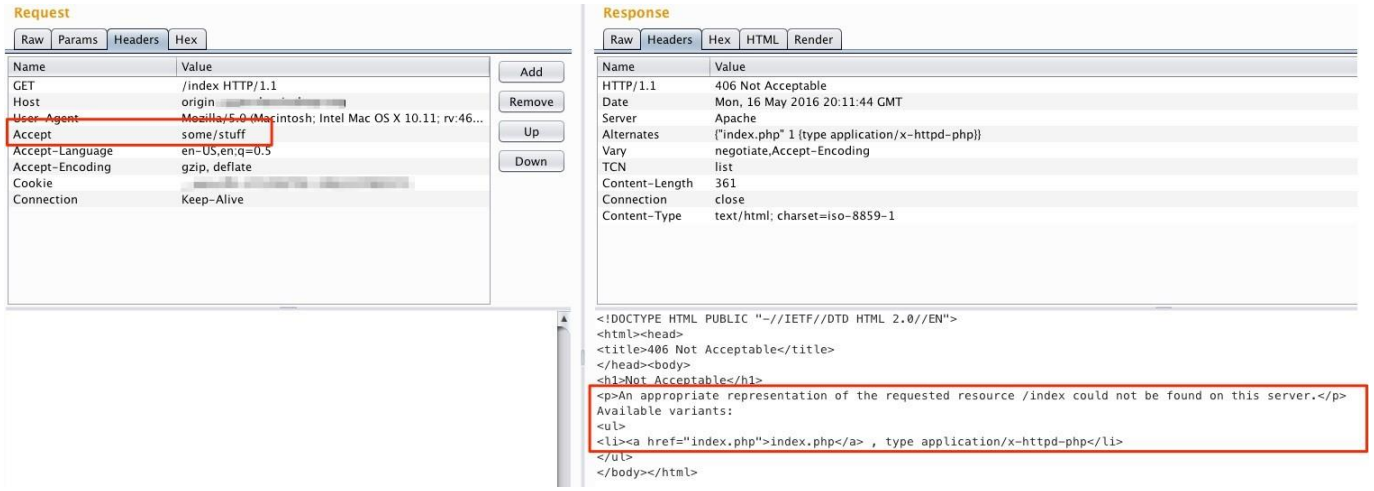
1

Status:

Not Remediated

Finding(s)

Evidence:



Evidence notes:
The left-hand side of the image shows a request for the file "index." The Right-hand side shows the server response telling us index does not exist, but index.php does.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

Low (1.25) = (Likelihood (1 + 1) /2 = Low (1) + Impact (2 + 1) /2 = Low (1.5)) /2

Reference(s):

http://www.tenable.com/plugins/index.php?view=single&id=10704
https://httpd.apache.org/docs/2.4/mod/mod_negotiation.html

CVSS:

(AV:N/AC:L/Au:N/C:P/I:N/A:N)

[Back to Top]

Appendix A

Approach

MalwareCare network penetration test combines the results from industry-leading scanning tools withmanual testing to enumerate and validate vulnerabilities, configuration errors, and business logic flaws. In- depth manual application testing enables us to find what scanners often miss.

Web applications are particularly vulnerable to external attack given that they are inherently designed to be accessible to the Internet. While automated scanners check for known vulnerabilities, they are incapable of actually reporting on real business risk. Our web application security testing helps you lower your risk of data breach, improve productivity, protect your brand, and maximize the ROI from your web applications.

MalwareCare's network penetration test service utilizes a risk-based approach to manually identify critical application-centric vulnerabilities that exist on all in-scope applications.

Using this approach, MalwareCare's comprehensive approach covers the classes of vulnerabilities in the Open WebApplication Security Project (OWASP) Top 10 2013 and beyond:

1. Injection (i.e.: SQL injection)
2. Broken Authentication and Session Management
3. Cross-site Scripting (XSS)
4. Insecure Direct Object Access
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross-site Request Forgery (CSRF)
9. Using Components with Known Vulnerabilities
10. Unvalidated Redirects and Forwards

Automated vs Manual Testing

MalwareCare's approach consists of about 80% manual testing and about 20% automated testing - actual results may vary slightly. While automated testing enables efficiency, it is effective in providing efficiency only during the initial phases of a penetration test. At MalwareCare, it is our belief that an effective and comprehensive test can only be realized through rigorous manual testing techniques.

Tools

In order to perform a comprehensive real-world assessment, MalwareCare utilizes commercial tools, internally developed tools and the same tools that hacker use on each and every assessment. Once again, our intent is to assess systems by simulating a real-world attack and we leverage the many tools at our disposal toeffectively carry out that task.

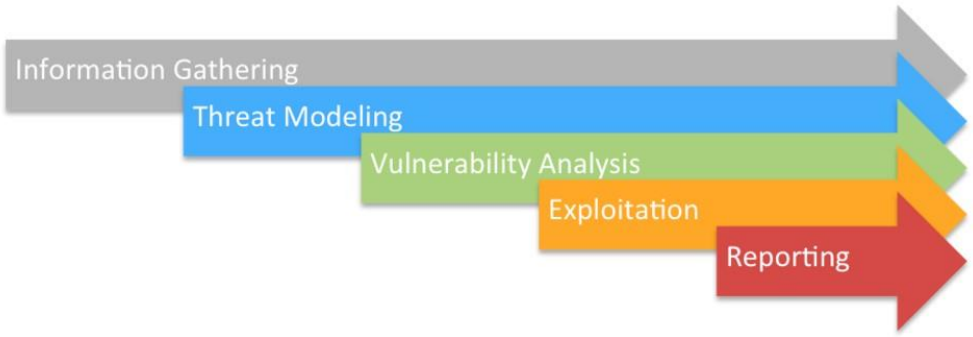
We make use of tools from the following categories (not a complete list):

- Commercial tools (i.e.: Burp Suite Pro, AppScan, WebInspect)
- Open source / Hacker tools (i.e.: Metasploit, BEeF, Kali Linux, OWASP Zap)
- MalwareCare developed tools (i.e.: nmapcli, Metasploit modules, PlugBot, various scripts)

Appendix A

Methodology

Penetration Testing Methodology



Information Gathering

The information-gathering phase consists of Google search engine reconnaissance, server fingerprinting, application enumeration and more. Information gathering efforts results in a compiled list of metadata and raw output with the goal of obtaining as much information about the application's makeup as possible. Reconnaissance includes initial domain foot printing, metafile leakage review, service enumeration and operating system and application fingerprinting. The purpose of this step is to collectively map the in-scope environment and prepare for threat identification.
During this phase, MalwareCare will perform the following:

- Use discovery tools to passively uncover information about the application (ie: robots.txt)
- Identify entry points into the application, such as administration portals or backdoors
- Perform application fingerprinting, in order to identify the use of a CMS (ie: Drupal) and the underlying dev language
- Send fuzzing requests to be used in the analysis of error codes that may disclose valuable information that could be used to launch a more targeted attack
- Actively scan for open services and develop a test plan for latter phases in the assessment

Threat Modeling

With the information collected from the previous step, security testing transitions to identifying vulnerabilities in the application. This typically begins with automated scans (i.e.: AppScan) initially but quickly morphs into manual testing techniques using more pointed and direct tools. During the threat-modeling step, assets are identified and categorized into threat categories. These may involve: sensitive documents, trade secrets, financial information, etc.
During this phase, MalwareCare will perform the following:

- Use open source, commercial and internally developed tools to identify well-known vulnerabilities (ie: AppScan, BURP, WebInspect, Metasploit)
- Spider the in-scope application(s) to effectively build a map of each of the features, components and areas of interest
- Use discovered sections, features, capabilities to establish threat categories to be used for more manual/rigorous testing (ie: file uploads, admin backdoors, web services, WYSIWYG editors)

Appendix A

- Send fuzzing requests to be used in the analysis of error codes that may disclose valuable information that could be used to launch a more targeted attack
- Build the application's threat model using the information gathered in this phase. This model will be used as a plan of attack for latter phases in the assessment.

Vulnerability Analysis

The vulnerability analysis step involves the documenting and analysis of vulnerabilities discovered as a result of the previous steps. This includes the analysis of out from the various security tools and manual testing techniques.
During this phase, MalwareCare will perform the following:

- Compile the list of areas of interest and develop a plan for exploitation
- Search and gather known exploits from various sources (ExploitDB, Pastebin, etc)
- Analyze the impact and likelihood for each potential exploitable vulnerability
- Select the best method and tools for properly exploiting each of the suspected exploitable vulnerabilities

Exploitation

Unlike a vulnerability assessment, a penetration test takes such a test quite a bit further by way of exploitation. Exploitation involves establishing access to application through the bypassing and exploitation of security controls in order to determine their actual real world risk. Throughout this step, we perform several manual tests incapable of being performed through automated means, such as scanners. During a MalwareCare penetration test, this phase consists of heavy manual testing tactics and is often the most time-intensive phase. Exploitation may include, but is not limited to: buffer overflow, SQL injection, OS commanding, cross-site scripting and more.

During this phase, MalwareCare will perform the following:

- Using the identified vulnerabilities in the previous phase, MalwareCare will manually exploit any identifiedvulnerabilities in order to validate them
- Capture and log evidence to provide proof of exploitation (ie: images, movies, screenshots, configs, etc.)
- Notify the client of any Critical or High findings upon discovery by telephone and email
- Upload validated exploits and their corresponding evidence/information to the project portal for client review
- Perform re-testing, per client request

Reporting

The reporting step is intended to deliver, rank and prioritize findings and generate a clear and actionable report, complete with evidence, to the project stakeholders. The presentation of findings can occur via Webex or in-person - whichever format is most conducive for communicating results.

Appendix A

During this phase, MalwareCare will perform the following:

- Ensure all findings have been uploaded to the project portal for client review
- Create the penetration test report, along with evidence, and upload it to the client portal for review
- Schedule a meeting with the client in an effort to present and talk through each of the identified vulnerabilities
- Optionally, additional meeting may take place to ensure the client understands the findings and recommendations for mitigation

Comprehensive Methodology

Each and every internal penetration test is conducted consistently using globally accepted and industry standard frameworks. In order to ensure a sound and comprehensive penetration test, MalwareCare leverages industry standard frameworks as a foundation for carrying out penetration tests. The underlying framework is based on the Open Web Application Security Project (OWASP).

OWASP is a globally accepted framework designed to enable the execution of effective penetration testing consistent with best practice all while ensuring a holistic and comprehensive evaluation. At MalwareCare,we consider this phase to be the most important and we take great care to ensure we've communicated the value of our service and findings thoroughly.

Appendix A

Risk Rating Overview

MalwareCare has adopted an industry-standard approach to assigning risk ratings to vulnerabilities. This approach is used in all our assessments and provides our clients with risk ratings that take into account a number of factors ranging from: Skill Level, Motive, Ease of Exploit, Loss of Integrity to Privacy/Reputational Damage.

Our comprehensive approach ensures that our clients' vulnerabilities are represented by their true real-world likelihood and potential impact to their business.

Appendix A

Risk Calculation

Risk Calculation is carried out through a quantitative method. The calculation is an industry standard approach and is widely adopted by many organizations across the globe. Please see the detail below for a walkthrough of the risk calculation process.

Calculation of *Likelihood* is achieved by the equation:

AVERAGE(Threat Agent + Vulnerability Factors) = Likelihood

Calculation of *Impact* is achieved by the equation:

AVERAGE(Technical Impact + Business Impact) = Impact

Calculation of the finding's overall *Risk Rating* is achieved by the following equation:

AVERAGE(Likelihood + Impact) = Risk Rating

Factors Explained

THREAT AGENT FACTORS

Factors in this category aid in establishing the real-world likelihood of exploitation. Overall, these factors take into account the knowledge and breadth of the threat.

- Skill Level – How technically skilled are the group of agents
- Motive – How motivated are the group of agents
- Opportunity – What resources/opportunity are required to find/exploit
- Size – How large is the group of agents

Appendix A

VULNERABILITY FACTORS

Factors in this category aid in establishing the real-world likelihood of exploitation. Overall, these factors take into account the ease of exploitation and how well known it might be.

- Ease of Discovery – How easy is it to discover this vulnerability
- Ease of Exploit – How easy is it to actually exploit this vulnerability
- Awareness – How well known is this vulnerability
- Intrusion Detection – How likely is this to be exploited

TECHNICAL IMPACT FACTORS

Factors in this category aid in establishing the estimated impact. Overall, these factors account for potential damage to CIA (Confidentiality, Integrity, Availability) with respect to data.

- Loss of Confidentiality – How much data could be disclosed and how sensitive
- Loss of Integrity – How much data could be corrupted/damaged
- Loss of Availability – How much service could be lost and how vital is it
- Loss of Accountability – Are the threat agents' action traceable to an individual

BUSINESS IMPACT FACTORS

Factors in this category aid in establishing the estimated impact. Overall, these factors account for potential damage to the business, such as reputation, finances and privacy.

- Financial Damage – How much financial damage would result
- Reputational Damage – Would an exploit cause reputational damage
- Non-Compliance – How much does exposure does non-compliance introduce
- Privacy Violation – How much personally identifiable information could be disclosed

Appendix A

Tools

Shown below is a list of the most commonly used tools during such an engagement. MalwareCare utilizes commercial, open source and MalwareCare-developed tools. Be advised this is not an completedand exhaustive list.

Nessus	nmap
Kali Linux	Wireshark
Metasploit	nmapcli
PlugBot	John the Ripper
Hydra	Nikto
OpenVAS	Cain & Abel
Olly Debugger	IDA Pro
hping	onesixtyone
AppScan	WebInspect
Hydra	Burp Suite Pro
Firewalk	fragroute / fragrouter
sqlmap	netifera
sslscan	Forify SCA
scapy	Mantra
TOR	Ethereal
sslscan	Forify SCA
i2p	tcpdump
OWASP ZAP	Aircrack
BeEF Framework	OWASP Xenotix XSS Exploit Framework
Spike	Cookiedigger
Paros Proxy	dsniff
Brutus	P0f
Kismet	dnsenum
Maltego	Skipfish
Social Engineering Toolkit	Armitage