**CSS300-1901B-01: Vulnerability Assessment and Management**

**Vulnerability Assessment and Management Guidelines Document**

**Student ABC**

**3/14/19**

Table of Contents

[Intrusion Tools and Techniques 3](#_Toc3463601)

[Auditing 3](#_Toc3463602)

[Audit Data Review 3](#_Toc3463603)

[Intrusion Detection 4](#_Toc3463604)

[Common Vulnerabilities and Exposures 6](#_Toc3463605)

[Definition of CVE and databases 6](#_Toc3463606)

[Calculation of CVSS 7](#_Toc3463607)

[Base and Temporal Metrics 8](#_Toc3463608)

[The use of the National Vulnerability Database 10](#_Toc3463609)

[CVSS Vulnerabilities 11](#_Toc3463610)

[Attack Methods 14](#_Toc3463611)

[Active and Passive attack 14](#_Toc3463612)

[Authenticated and Unauthenticated 21](#_Toc3463613)

[Intrusion detection System Policies 23](#_Toc3463614)

[Physical and Environmental security 23](#_Toc3463615)

[Human Resource security 25](#_Toc3463616)

[Access control 25](#_Toc3463617)

[Protective Measures 27](#_Toc3463618)

[Tool Description 27](#_Toc3463619)

[List of Identified Hosts 27](#_Toc3463620)

[Scaling of our Threats 29](#_Toc3463621)

[False Positive Information 29](#_Toc3463622)

[Potential Safeguards and Remediation 30](#_Toc3463623)

[Bibliography 31](#_Toc3463624)

## Intrusion Tools and Techniques

### Auditing

A data audit refers to the auditing of data to assess its quality or utility for a specific purpose. Auditing data, unlike auditing finances, involves looking at key metrics, other than quantity, to create conclusions about the properties of a data set.

Audit data is a data log of action performed in a given digital environment; which would be useful in detecting data breaches and tracking activity within the framework that is being implemented.

Audit data should be stored in secured private servers or private networks, since the data itself may usually be sensitive in nature.

### Audit Data Review

At the heart of most devices that provide protection for IT networks is an ability to log events and take actions based on those events. The IT networks need to be protected and there for the data form devices associated with these networks should be considered for auditing. This [application and system monitoring](https://searchwindowsserver.techtarget.com/answer/What-kind-of-log-analytics-does-Microsoft-OMS-perform) provides details both on what has happened to the device and what is happening. It provides security against lapses in perimeter and application defenses by alerting you to problems so defensive measures can be taken before any real damage is done. Without monitoring, you have little chance of discovering whether a live application is being attacked or has been compromised.

Critical applications, processes handling valuable or sensitive information, previously compromised or abused systems, and systems connected to third parties or the Internet all [require active monitoring](https://searchvmware.techtarget.com/tip/Identify-abnormal-application-behavior-with-VMware-AppDefense). Any seriously suspicious behavior or critical events must generate an alert that is assessed and acted on. Although you will need to carry out a risk assessment for each application or system to determine what level of [audit, log review](http://searchsecurity.techtarget.co.uk/answer/How-to-manage-logs) and monitoring is necessary, you will need to log at least the following:

* User IDs
* Date and time of log on and log off, and other key events
* Terminal identity
* Successful and failed attempts to access systems, data or applications
* Files and networks accessed
* Changes to system configurations
* Use of system utilities
* Exceptions and other security-related events, such as alarms triggered
* Activation of protection systems, such as intrusion detection systems and antimalware

### Intrusion Detection

Collecting this data will assist in [access control monitoring](https://searchdisasterrecovery.techtarget.com/tip/Six-ITGC-audit-controls-to-improve-business-continuity) and can provide audit trails when investigating an incident. While most logs are covered by some form of regulation these days and should be kept as long as the requirements call for, any that are not should be kept for a minimum period of one year, in case they are needed for an investigation.  However, monitoring must be carried out in line with relevant legislation, which in the UK is the [Regulation of Investigatory Powers and Human Rights Acts](http://searchsecurity.techtarget.co.uk/definition/Regulation-of-Investigatory-Powers-Act). Employees should be made aware of your monitoring activities in the network acceptable use policy. No matter how extensive your logging, [log files](https://searchwindowsserver.techtarget.com/tutorial/PowerShell-logging-boosts-security-in-the-enterprise) are worthless if you cannot trust their integrity.

Log files are a great source of information only if you review them. Simply purchasing and deploying a [log management product](https://searchdatacenter.techtarget.com/tip/Open-source-log-management-options-for-Linux-distributions) won’t provide any additional security. You have to use the information collected and analyze it on a regular basis; for a high-risk application, this could mean automated reviews on an hourly basis. ISO/IEC 27001 control A.10.10.2 not only requires procedures for monitoring the use of information processing facilities, but demands the results are reviewed regularly to identify possible security threats and incidents.

However, even small networks can generate too much information to be analyzed manually. This is where log analyzers come in, as they automate the [auditing and analysis of logs](https://searchsecurity.techtarget.com/answer/Audit-log-security-How-to-monitor-and-protect-audit-logs), telling you what has happened or is happening, and revealing unauthorized activity or abnormal behavior. This feedback can be used to improve IDS signatures or firewall rule sets. Such improvements are an iterative process, as regularly tuning your devices to maximize their accuracy in recognizing true threats will help reduce the number of false positives. Completely eliminating false positives, while still maintaining strict controls, is next to impossible, particularly as new threats and changes in the network structure will affect the effectiveness of existing rule sets. Log analysis can also provide a basis for focused security awareness training, reduced network misuse and stronger policy enforcement.

The data being secure and accurate is based on whether that data is coming from a trusted system. A trusted system is one that has been shown to warrant some degree of trust that it will perform certain activities faithfully, that is, in accordance with users’ expectations. Contrary to popular usage, “trusted” in this context does not mean hope, in the sense of,” I hope this system protects me form malicious code.” Hope is trust with little justification; trusted systems have much evidence to justify users’ trust.

Trusted systems have three characteristics:

* A defined policy that details what security qualities it enforces
* Appropriate measures and mechanisms by which it can enforce that security adequately
* Independent scrutiny or evaluation to ensure that the mechanisms have been selected and implemented properly so that the security policy is in fact enforced.

## Common Vulnerabilities and Exposures

### Definition of CVE and databases

The Common Vulnerabilities and Exposures (CVE) system provides a reference-method for publicly known information-security vulnerabilities and exposures. The National Cybersecurity FFRDC, operated by the Mitre Corporation, maintains the system, with funding from the National Cyber Security Division of the United States Department of Homeland Security.

The Security Content Automation Protocol uses CVE, and CVE IDs are listed on MITRE's system as well as in the US National Vulnerability Database.

CVE identifiers are intended for use with respect to identifying vulnerabilities:

Common Vulnerabilities and Exposures (CVE) is a dictionary of common names (i.e., CVE Identifiers) for publicly known information security vulnerabilities. CVE’s common identifiers make it easier to share data across separate network security databases and tools and provide a baseline for evaluating the coverage of an organization’s security tools. If a report from one of your security tools incorporates CVE Identifiers, you may then quickly and accurately access fix information in one or more separate CVE-compatible databases to remediate the problem.

Users who have been assigned a CVE identifier for a vulnerability are encouraged to ensure that they place the identifier in any related security reports, web pages, emails, and so on.

### Calculation of CVSS

The Common Vulnerability Scoring System (CVSS) provides a way to capture the principal characteristics of a vulnerability, and produce a numerical score reflecting its severity, as well as a textual representation of that score. The numerical score can then be translated into a qualitative representation (such as low, medium, high, and critical) to help organizations properly assess and prioritize their vulnerability management processes.

CVSS affords three important benefits:

It provides standardized vulnerability scores. When an organization uses a common algorithm for scoring vulnerabilities across all IT platforms, it can leverage a single vulnerability management policy defining the maximum allowable time to validate and remediate a given vulnerability.

It provides an open framework. Users may be confused when a vulnerability is assigned an arbitrary score by a third party. With CVSS, the individual characteristics used to derive a score are transparent.

CVSS helps prioritize risk. When the environmental score is computed, the vulnerability becomes contextual to each organization, and helps provide a better understanding of the risk posed by a vulnerability to the organization.

Since its initial release in 2004, CVSS has enjoyed widespread adoption. In September 2007, CVSS v2.0 was adopted as part of the Payment Card Industry Data Security Standard (PCI DSS). To comply with PCI DSS, merchants processing credit cards must demonstrate that none of their computing systems has a vulnerability with a CVSS score greater than or equal to 4.0. In 2007 NIST included CVSS v2.0 as part of their Security Content Automation Protocol (SCAP). In April 2011, CVSS v2.0 was formally adopted as an international standard for scoring vulnerabilities (ITU-T X.1521).

### Base and Temporal Metrics

The Temporal metric group reflects the characteristics of a vulnerability that may change over time but not across user environments. For example, the presence of a simple-to-use exploit kit would increase the CVSS score, while the creation of an official patch would decrease it. The Temporal metrics measure the current state of exploit techniques or code availability, the existence of any patches or workarounds, or the confidence that one has in the description of a vulnerability.

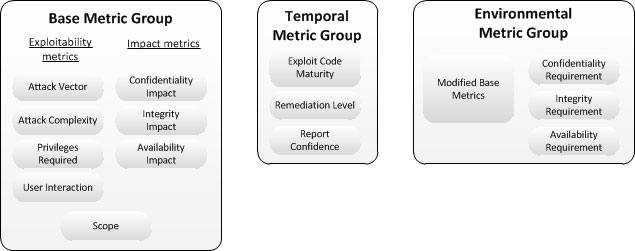
The Base metric group represents the intrinsic characteristics of a vulnerability that are constant over time and across user environments. It is composed of two sets of metrics: The Exploitability metrics and the Impact metrics.

As mentioned, the Exploitability metrics reflect the characteristics of the thing that is vulnerable, which we refer to formally as the vulnerable component. Therefore, each of the Exploitability metrics listed below should be scored relative to the vulnerable component and reflect the properties of the vulnerability that lead to a successful attack.

The Impact metrics refer to the properties of the impacted component. Whether a successfully exploited vulnerability affects one or more components, the impact metrics are scored according to the component that suffers the worst outcome that is most directly and predictably associated with a successful attack. That is, analysts should constrain impacts to a reasonable, final outcome which they are confident an attacker is able to achieve.

If a scope change has not occurred, the Impact metrics should reflect the confidentiality, integrity, and availability (CIA) impact to the vulnerable component. However, if a scope change has occurred, then the Impact metrics should reflect the CIA impact to either the vulnerable component, or the impacted component, whichever suffers the most severe outcome.

The difference between the two is that Base are constant over time for user environments while Temporal is not constant over time for user environments.



### The use of the National Vulnerability Database

The NVD is the U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP). This data enables automation of vulnerability management, security measurement, and compliance. The NVD includes databases of security checklist references, security related software flaws, misconfigurations, product names, and impact metrics.

Originally created in 2000 (called Internet - Categorization of Attacks Toolkit or ICAT), the NVD has undergone multiple iterations and improvements and will continue to do so to deliver its services. The NVD is a product of the NIST Computer Security Division, Information Technology Laboratory and is sponsored by the Department of Homeland Security’s National Cyber Security Division.

The NVD performs analysis on CVEs that have been published to the CVE Dictionary. NVD staff are tasked with analysis of CVEs by aggregating data points from the description, references supplied and any supplemental data that can be found publicly at the time. This analysis results in association impact metrics (Common Vulnerability Scoring System - CVSS), vulnerability types (Common Weakness Enumeration - CWE), and applicability statements (Common Platform Enumeration - CPE), as well as other pertinent metadata. The NVD does not actively perform vulnerability testing, relying on vendors, third party security researchers and vulnerability coordinators to provide information that is then used to assign these attributes. As additional information becomes available CVSS scores, CWEs, and applicability statements are subject to change. The NVD endeavors to re-analyze CVEs that have been amended as time and resources allow to ensure that the information offered is up to date.

### CVSS Vulnerabilities

The Common Vulnerability Scoring System (CVSS) is a [framework](https://whatis.techtarget.com/definition/framework) for rating the severity of security vulnerabilities in software. CVSS attempts to assign severity scores to vulnerabilities, allowing responders to prioritize responses and resources according to threat. Operated by the Forum of Incident Response and Security Teams (FIRST), the CVSS uses an algorithm to determine three severity rating scores: Base, Temporal and Environmental. The scores are numeric; they range from 0.0 through 10.0 with 10.0 being the most severe.

The Base score is the metric most relied upon by enterprises and deals with the inherent qualities of [a vulnerability](https://searchsecurity.techtarget.com/definition/vulnerability-assessment-vulnerability-analysis). The Temporal scores represent the qualities of the vulnerability that change over time, and the Environmental score represents the qualities of the vulnerability that are specific to the affected user's environment. According to the most recent version of the CVSS, v3.0, a score of 0.0 receives a "None" rating; a 0.1-3.9 score gets a "Low" severity rating; a score of 4.0-6.9 is a "Medium" rating; score of 7.0-8.9 is a "High" rating; and a score of 9.0 - 10.0 is a "Critical" rating.

The CVSS allows organizations to prioritize which vulnerabilities to fix first and gauge the impact of the vulnerabilities on their systems. Many organizations use the CVSS, and the [National Vulnerability Database](https://whatis.techtarget.com/definition/National-Vulnerability-Database-NVD) provides scores for most known vulnerabilities. According to the NVD, a CVSS base score of 0.0-3.9 is considered "Low" severity; a base CVSS score of 4.0-6.9 is "Medium" severity; and base score of 7.0-10.0 is "High" severity.

The CVSS was introduced in 2005 by the National Infrastructure Advisory Council (NIAC), which turned over management and development of the standard to FIRST. The current version, CVSS 3.0, was introduced in June of 2015. As a free and open standard, several vendors such as Oracle have customized their own versions of the CVSS.

Three vulnerabilities that could potentially exist in your organization, that have a CVSS severity rated as high would include threats to confidentiality, integrity and availability.

**Confidentiality**

The confidentiality (C) metric describes the impact on the confidentiality of data processed by the system.

|  |  |  |
| --- | --- | --- |
| **Value** | **Description** | **Score** |
| None (N) | There is no impact on the confidentiality of the system. | 0.0 |
| Partial (P) | There is considerable disclosure of information, but the scope of the loss is constrained such that not all of the data is available. | 0.275 |
| Complete (C) | There is total information disclosure, providing access to any / all data on the system. Alternatively, access to only some restricted information is obtained, but the disclosed information presents a direct, serious impact. | 0.660 |

**Integrity**

The Integrity (I) metric describes the impact on the integrity of the exploited system.

|  |  |  |
| --- | --- | --- |
| **Value** | **Description** | **Score** |
| None (N) | There is no impact on the integrity of the system. | 0.0 |
| Partial (P) | Modification of some data or system files is possible, but the scope of the modification is limited. | 0.275 |
| Complete (C) | There is total loss of integrity; the attacker can modify any files or information on the target system. | 0.660 |

**Availability**

The availability (A) metric describes the impact on the availability of the target system. Attacks that consume network bandwidth, processor cycles, memory or any other resources affect the availability of a system.

|  |  |  |
| --- | --- | --- |
| **Value** | **Description** | **Score** |
| None (N) | There is no impact on the availability of the system. | 0.0 |
| Partial (P) | There is reduced performance or loss of some functionality. | 0.275 |
| Complete (C) | There is total loss of availability of the attacked resource. | 0.660 |

## Attack Methods

### Active and Passive attack

#### Passive attacks

The main types of passive attacks are traffic analysis and release of message contents. During a traffic analysis attack, the eavesdropper analyzes the traffic, determines the location, identifies communicating hosts and observes the frequency and length of exchanged messages. He uses all this information to predict the nature of communication. All incoming and outgoing traffic of the network is analyzed, but not altered.

WINDOWS:

Microsoft Message Analyzer tool, the successor to Microsoft Network Monitor 3.4, has an intuitive and flexible UI with effective filtering options that allow you to break down and drill into captured packets (or ‘messages’ as they are called in Message Analyzer). By adding ‘Color Rules’ to different protocol traffic, you can make scanning through areas of interest easier and faster.

Some of its highlighted features include automated data capture (using PowerShell cmdlets to start or stop traces based on a particular trigger), TLS/SSL decryption support and customizable filter expressions.

Microsoft Message Analyzer allows you to assess multiple log data sources from a single pane of glass. You can capture, view and analyze network protocol traffic side-by-side with other system or application events (e.g., Event Logs or SQL Tables), making it a valuable addition to your network toolkit.

For a release of message content, a telephonic conversation, an E-mail message or a transferred file may contain confidential data. A passive attack monitors the contents of the transmitted data.

Passive attacks are very difficult to detect because they do not involve any alteration of the data. When the messages are exchanged neither the sender nor the receiver is aware that a third party may capture the messages. This can be prevented by encryption of data.

UNIX:

An intrusion detection system capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching. **Snort** can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and more. Snort uses a flexible rules language to describe traffic it should collect or pass, and includes a detection engine using a modular plug-in architecture. Snort has real-time alerting capability as well, incorporating alerting mechanisms for Syslog, user-specified files, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient. Snort has three primary uses: as a straight packet sniffer like tcpdump, as a packet logger that is useful for network traffic debugging, and as a full-blown network intrusion detection system.

#### Active Attacks

In general, a masquerade is a disguise. In terms of communications security issues, a masquerade is a type of attack where the attacker pretends to be an authorized user of a system in order to gain access to it or to gain greater privileges than they are authorized for. A masquerade may be attempted using stolen logon IDs and passwords, through finding security gaps in programs, or through bypassing the authentication mechanism. The attempt may come from within an organization, for example, from an employee; or from an outside user through some connection to the public network. Weak authentication provides one of the easiest points of entry for a masquerade, since it makes it much easier for an attacker to gain access. Once the attacker has been authorized for entry, they may have full access to the organization's critical data, and (depending on the privilege level they pretend to have) may be able to modify and delete software and data and make changes to network configuration and routing information. (Rouse, 2007)

Windows:

In another variation of this technique, an adversary may use a renamed copy of a legitimate utility, such as rundll32.exe. An alternative case occurs when a legitimate utility is moved to a different directory and also renamed to avoid detections based on system utilities executing from non-standard paths.

An example of abuse of trusted locations in Windows would be the C:\Windows\System32 directory. Examples of trusted binary names that can be given to malicious binares include "explorer.exe" and "svchost.exe".

Linux:

Another variation of this technique includes malicious binaries changing the name of their running process to that of a trusted or benign process, after they have been launched as opposed to before.

An example of abuse of trusted locations in Linux would be the /bin directory. Examples of trusted binary names that can be given to malicious binares include "rsyncd" and "dbus-inotifier".

*Replay*

A **replay attack** (also known as **playback attack**) is a form of [network](https://en.wikipedia.org/wiki/Computer_network) attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an [adversary](https://en.wikipedia.org/wiki/Adversary_(cryptography)) who intercepts the data and re-transmits it, possibly as part of a [masquerade attack](https://en.wikipedia.org/wiki/Spoofing_attack) by [IP](https://en.wikipedia.org/wiki/Internet_Protocol) [packet](https://en.wikipedia.org/wiki/Packet_(information_technology)) substitution. This is one of the lower tier versions of a "[Man-in-the-middle attack](https://en.wikipedia.org/wiki/Man-in-the-middle_attack)".

Another way of describing such an attack is: "an attack on a security protocol using replay of messages from a different context into the intended (or original and expected) context, thereby fooling the honest participant(s) into thinking they have successfully completed the protocol run."

Windows:

A replay attack occurs when an intruder steals a packet from the network and forwards that packet to a service or application as if the intruder was the user who originally sent the packet. When the packet is an authentication packet, the intruder can use the replay attack to authenticate on another person's behalf and consequently access that person's resources or data.

To protect against replay attacks, the Kerberos authentication protocol uses the concept of an authenticator. A Kerberos authenticator is embedded in the Kerberos protocol exchanges that occur between the authenticating client and the authentication server (in Windows, the domain controller—DC). It holds additional authentication data, such as the ticket lifetime, and most important, the client's timestamp. When the Kerberos logic on a DC or resource server validates a Kerberos authentication message, it will always check the authenticator's timestamp. If the timestamp is earlier or the same as a previous authenticator, the server-side Kerberos logic will reject the packet because it considers it part of a replay attack and user authentication will fail. The Kerberos server-side logic also compares the timestamp in the authenticator to the local server time. If the timestamp in the authenticator isn't within five minutes of the time on the server, it will also reject the packet. These five minutes are referred to as the Kerberos time skew. In Windows, the time skew defaults to five, but you can change this value if desired. To do so, you use the Maximum tolerance for computer clock synchronization [Group Policy Object](http://windowsitpro.com/systems-management/active-directory-group-policy) (GPO) setting located in the GPO folder Computer Configuration\Windows Settings\Security Settings\Account Policies\Kerberos Policy.

All this explains why it has become very important to keep the computer clocks synchronized in a [Windows Active Directory](http://windowsitpro.com/systems-management/active-directory-group-policy) (AD) forest, starting with Windows 2000 (when Kerberos became the default Windows authentication protocol). For that purpose, Windows includes the Windows Time service, which is crucial to the proper functioning of the Kerberos authentication services. To keep the system clocks on all computers in a Windows domain within five minutes, the Windows Time service uses the Network Time Protocol (NTP). OSs prior to Windows Server 2003 use the Simple Network Time Protocol (SNTP), which is the predecessor of NTP.

*Modification of Message*

In a message modification attack, an intruder alters packet header addresses to direct a message to a different destination or modify the data on a target machine.

UNIX:

A Problem in Establishing the Trusted Computing Base

In the paper "Reflections on Trusting Trust" by Ken Thompson, one of the original developers of UNIX at Bell Labs, details a major flaw in breaking into the trusted code. Let's say he wanted to have some special code that would allow a very specific username to be able to login to any machine by modifying "login.c" and distributing it as an update:

if (strcmp(user, "ken") == 0) {

uid = 0;

login();

}

As we can see, this would allow anyone who knew this secret username "ken" to login to any UNIX computer and act as the root. But UNIX is open source, so surely someone would catch it! Ken Thompson took it one step further.

He proposed that he could hypothetically modify the C compiler itself to specifically generate the strcmp function if compiling the file "login.c". So now "login.c" looks safe to anyone who reviews it. But now, someone could look at the C compiler source code and discover that trickery as well. Ken Thompson could cover this up too by programming the compiler for the compiler, so that (if compiling the C compiler), generate the code that would modify the C compiler to generate the code that would modify "login.c". Now we run into trouble, and there is no way to verify that this has not been done already in any of the operating systems in use today including Ubuntu, Redhat, Windows 8, etc. In a sense, we have to not only trust the trusted computing base, but also trust that the developers were honest to not conceal something like that if we wish to use their operating system.

*Denial of Service:*

It prevents normal use of communication facilities. This attack may have a specific target. For example, an entity may suppress all messages directed to a particular destination. Another form of service denial is the disruption of an entire network wither by disabling the network or by overloading it by messages so as to degrade performance.

In a distributed denial-of-service ([DDoS](https://searchsecurity.techtarget.com/definition/distributed-denial-of-service-attack)) exploit, large numbers of compromised systems (sometimes called a botnet or zombie army) attack a single target.

UNIX:

There are literally hundreds of DoS and DDoS tools available. Within [Kali](https://null-byte.wonderhowto.com/how-to/hack-like-pro-getting-started-with-kali-your-new-hacking-system-0151631/), we can find auxiliary modules within Metasploit specifically for DoSing. If we navigate to:

**kali > cd /usr/share/metasplot-framework/auxiliary/dos**

And list the contents of that directory, we can see that Metasploit has organized its DoS tools by the type of target. There are hundreds of denial-of-service tools in Metasploit.

We can also find hundreds of DoS tools in the [Exploit Database](https://null-byte.wonderhowto.com/how-to/hack-like-pro-find-exploits-using-exploit-database-kali-0156399/) built into Kali, and still more at [Exploit-DB.com](http://www.exploit-db.com). We can find a listing of the Exploit-DB DoS tools by navigating to:

**kali > /usr/share/exploitdb/platforms/windows/dos**

A long listing (**ls -l**) of this directory lists all of the Windows DoS tools. A similar, shorter list is at **/usr/share/exploitdb/platforms/Linux/dos**.

### Authenticated and Unauthenticated

Data protection in any system is very essential. It protects the unauthorized access to information. However, cases of attack have been on the rise. This can take the form of authenticated and unauthenticated attacks. **Authentication** is essential in providing security to web applications. Authenticated attack targets the authentication process of a website. The attack exploits the site that one uses to verify his or her identity while accessing an application or a service. An authenticated attack is usually accomplished by someone who is trusted (Endignoux & Vizár, 2017). The attacker gains access to the login credentials of an individual then login into the system then performs the attack. Those who are performing authenticated attack utilize the username and password of someone to prove their identity. The application then allows the user to gain access to the web application based on the credentials supplied. Some of the examples of authenticated attacks include brute force, insufficient authentication, and weak password recovery validation (Endignoux & Vizár, 2017). In brute force, the attacker is allowed to guess the login credentials such as the username, the password, cryptographic key and the credit card number of another person through the use of an automated process. The attacker utilizes trial and error to gain access to the web application. Insufficient authentication allows the attacker to gain access to a web application or site that contains valuable and sensitive information. The attacker is not necessarily required to authenticate the website. This attack is possible where there is no enough security to prevent unauthorized access to sensitive content. Weak password recovery validation allows the attacker to gain access to another person’s website. The hacker can then obtain, change and validate the password of the user.

On the other hand, the **unauthenticated attack** is the process of exploring the vulnerability of a network system without necessarily having to log in as an authorized user. It is usually carried out by malicious attackers who want to gain access to sensitive information of the other person without supplying valid login credentials. After login into the network, the attacker then acts as the trusted user of the website or the network system. In other words, the unauthenticated attack is where the unauthorized user circumvents the security of a network system by utilizing a vulnerability or flaw that does not entail or contain authentication (Kamarei, Patooghy & Fazeli, 2016). In authentication attack, the attacker is not required to pass through an authentication boundary to get access to the sensitive data and information. The wizards who get access to the sensitive information do have custom links that they use to go to the next step or the previous step. In this case, they will be able to confirm the data entered into the network system without the knowledge of the victim. The unauthenticated attack is typical in the shopping carts and contact forms (Kamarei, Patooghy & Fazeli, 2016). Most web shops allow users to fill the shopping carts before they are provided with the login credentials. The attacker can exploit this scenario by reading the products that the victim fills in the shopping cart without passing authentication boundary. Examples of unauthenticated attack include reconnaissance attack, nuisance attack, spam attack and phishing attack among others. Reconnaissance attack is where the hacker sends an invite message to the victim and determines the vulnerability of the SIP machine after obtaining feedback. A phishing attack is where the attacker poses as the authorized financial institution of the victim. The hacker modifies the invite message to get the personal credentials of the victim.

It is important to install modern authentication frameworks to prevent unauthorized access to the sensitive content of the user. Both authenticated and unauthenticated attacks can be avoided if the best applications are put in place to mitigate session fixation.

## Intrusion detection System Policies

The purpose of a policy is ensuring that all individuals with the organization understand their responsibilities in relation to incident response. It also gives guidelines for an organization’s members that will protect company assets and prevent loses during the operation of its functions.

This policy is for preventing not only malicious insider intrusion, but physical intrusion as well.

### Physical and Environmental security

* Physical access to premises and support infrastructure (communications, power, air conditioning etc.) must be monitored and restricted to prevent, detect and minimize the effects of unauthorized and inappropriate access, tampering, vandalism, criminal damage, theft etc. Usually unauthorized access to physical company materials may lead our or an affiliate organization being compromised.
* The list of people authorized to access secure areas must be reviewed and approved periodically (at least once a year) by Administration or Physical Security Department and cross-checked by their departmental managers. This will insure that we have approved members on our team who are trustworthy enough to handle data given to them.
* Photography or video recording is forbidden inside Restricted Areas without prior permission from the designated authority. This will protect the identity of members of our organization who value their privacy.
* Suitable video surveillance cameras must be located at all entrances and exits to the premises and other strategic points such as Restricted Areas, recorded and stored for at least one month, and monitored around the clock by trained personnel. This will give us the ability to have a preemptive response to possibly criminal activity or have video evidence if an incident occurs.
* Access cards permitting time-limited access to general and/or specific areas may be provided to trainees, vendors, consultants, third parties and other personnel who have been identified, authenticated, and authorized to access those areas other than in public areas such as the reception foyer, and private areas such as rest rooms, visitors should be escorted at all times by an employee while on the premises. Limiting the access of certain areas or files will keep them more secure.
* The date and time of entry and departure of visitors along with the purpose of visits must be recorded in a register maintained and controlled by Site Security or Reception. This would be also including an auditing of network activities.
* Everyone on site (employees and visitors) must wear and display their valid, always issued pass, and must present their pass for inspection on request by a manager, security guard or concerned employee.
* Access control systems must themselves be adequately secured against unauthorized/inappropriate access and other compromises.
* Fire/evacuation drills must be conducted periodically (at least once a year).
* Smoking is forbidden inside the premises other than in designated Smoking Zones.

### Human Resource security

* All employees must be screened prior to employment, including identity verification using a passport or similar photo ID and at least two satisfactory professional references. Additional checks are required for employees taking up trusted positions.
* All employees must formally accept a binding confidentiality or non-disclosure agreement concerning personal and proprietary information provided to or generated by them during employment. This will protect the employee and the company.
* Human Resources department must inform Administration, Finance and Operations when an employee is taken on, transferred, resigns, is suspended or released on long-term leave, or their employment is terminated.
* Upon receiving notification from HR that an employee's status has changed, Administration must update their physical access rights and IT Security Administration must update their logical access rights accordingly.
* An employee's manager must ensure that all access cards, keys, IT equipment, storage media and other valuable corporate assets are returned by the employee on or before their last day of employment.

### Access control

* User access to corporate IT systems, networks, applications and information must be controlled in accordance with access requirements specified by the relevant Information Asset Owners, normally according to the user's role.
* Generic or test IDs must not be created or enabled on production systems unless specifically authorized by the relevant Information Asset Owners.
* After a predefined number of unsuccessful logon attempts, security log entries and (where appropriate) security alerts must be generated and user accounts must be locked out as required by the relevant Information Asset Owners.
* Passwords or pass phrases must be lengthy and complex, consisting of a mix of letters, numerals and special characters that would be difficult to guess.
* Passwords or pass phrases must not be written down or stored in readable format.
* Authentication information such as passwords, security logs, security configurations and so forth must be adequately secured against unauthorized or inappropriate access, modification, corruption or loss.
* Privileged access rights typically required to administer, configure, manage, secure and monitor IT systems must be reviewed periodically (at least twice a year) by Information Security and cross-checked by the appropriate departmental managers.
* Users must either log off or password-lock their sessions before leaving them unattended.
* Password-protected screensavers with an inactivity timeout of no more than 10 minutes must be enabled on all workstations/PCs.
* Write access to removable media (USB drives, CD/DVD writers etc.) must be disabled on all desktops unless specifically authorized for legitimate business reasons.

## Protective Measures

### Tool Description

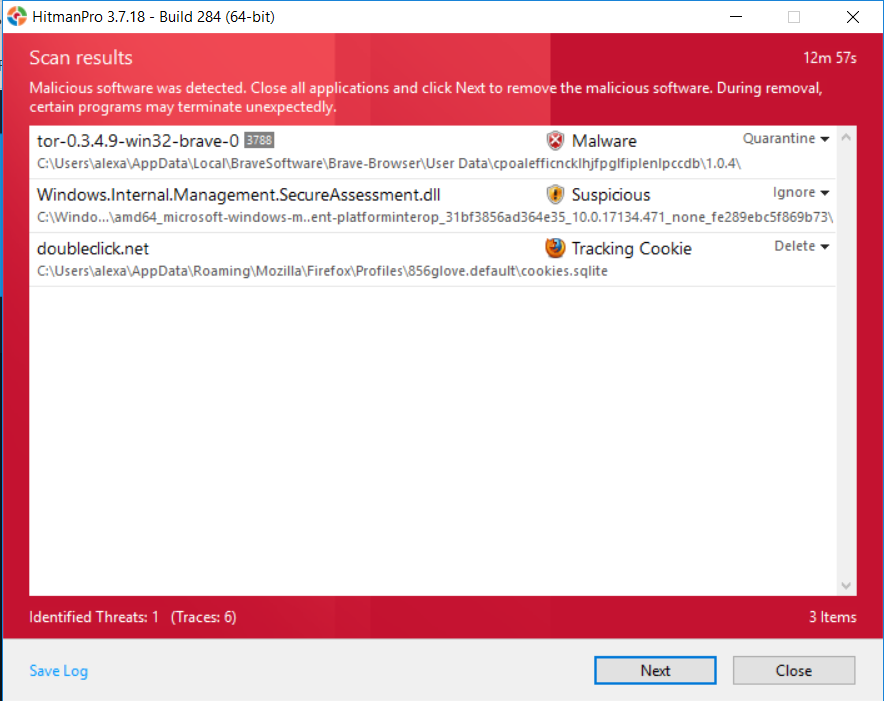
The tool I will be using to scan our System will be Hitman Pro.

HitmanPro (formerly Hitman Pro) is a portable [antimalware](https://en.wikipedia.org/wiki/Antivirus_software) program, which aims to detect and (if found) remove [malicious files](https://en.wikipedia.org/wiki/Malware) and [registry](https://en.wikipedia.org/wiki/Windows_registry) entries related to [rootkits](https://en.wikipedia.org/wiki/Rootkits), [trojans](https://en.wikipedia.org/wiki/Trojan_horse_(computing)), [viruses](https://en.wikipedia.org/wiki/Computer_virus), [worms](https://en.wikipedia.org/wiki/Computer_worm), [spyware](https://en.wikipedia.org/wiki/Spyware), [adware](https://en.wikipedia.org/wiki/Adware), [rogue antivirus programs](https://en.wikipedia.org/wiki/Rogue_security_software), [ransomware](https://en.wikipedia.org/wiki/Ransomware), and other [malware](https://en.wikipedia.org/wiki/Malware) from infected computers.

Suspicious objects are analyzed across an internet connection using a range of online malware detection services (see [multiscanning](https://en.wikipedia.org/wiki/Multiscanning" \o "Multiscanning)), and can be removed by HitmanPro if confirmed. Latest Version 3.7.9 uses [Bitdefender](https://en.wikipedia.org/wiki/Bitdefender) and [Kaspersky Lab](https://en.wikipedia.org/wiki/Kaspersky_Lab) as in-cloud technology partners. There is a very low system load because the virus definitions are not installed and tested in depth on the computer in question; rather they are analyzed remotely.

The company behind HitmanPro, Surfright, was acquired by [Sophos](https://en.wikipedia.org/wiki/Sophos) in December 2015.

### List of Identified Hosts



(example and list of a scan using HitmanPro)

This program doesn’t necessarily provide information on the IP addresses of host associated with each malware instance; however, the name and location of each threat within the operating system is displayed allowing for further investigation to be made in removing any threats to our network, at least on the system level. By scanning and removing any malware or trojans, we can create a deterrent to elements that would threaten our network from the inside.

### Scaling of our Threats

In this section and a later section, I will be rating the threat level of these threat from 1-5; 1 being high threat and 5 being low threat.

When rating the threat level of these identified threats on their severity to our system, we must consider the location of where this threat is located, and what would happen if the threat is removed. Our first threat is located within a browser and identified as malware (the tor-0.3 threat). Thankfully our program quarantined the threat. This is a high-level threat which I will give a 1, because it is a file lodged within a folder of the browser in question; meaning it could potentially be gathering and transmitting data in real-time.

Another threat would be the doubleclick.net cookie. Cookies are files that are placed on the system to give websites the ability to track your data and movement. By deleting cookies, you can better retain anonymity online; which is valuable in the security field and generally to keep your system safer. This cookie threat would get a 2 rating.

### False Positive Information

False positive information is data that a scanning tool reports as a threat when it’s not a threat or might even be an integral part of the operation of any given program or application. It can also be the reporting of a harmless entity attached to the system.

If I were to rate the threat with the extension of .dll presented from the screencapture with a score of 1-5, 1 being highest and 5 being lowest, I’d give it a 1. The reason being is that the threat is a .dll file, which means it’s situated as a component of the browser’s operation. Because I’m not a windows programmer I can’t determine the necessity of this file, and to delete this file could damage this computer. That is why before removing the file from the registry it is wise to backup any important files in case your computer becomes inoperable.

This file could be a false positive; I’d have to research what that .dll file does. Since it is a file within the system as a .dll, that may be giving Microsoft additional information on my network to help me; however, to be on the safe side any file that are deemed suspicious or malware should be considered for deletion with the carefulness that removing the file doesn’t damage the operating system. If not for this scanning program I wouldn’t have been able to know where the threat was lodged within my system.

### Potential Safeguards and Remediation

When a file is deleted through HitmanPro, and the program is still operational, I usually conclude that it isn’t needed, and I’ve made the system potentially safer by deleting the file. Researching what the cookies and trojans are that are found on your operating system does help to understand what and how to keep a system and network safe, and where these threats may be coming from to avoid them in the future.

A safeguard for threats to a network could include the use of Intrusion detection tools to defend against hackers. Regularly scanning one’s operating system with antimalware tools can better eliminate the risk of having malware running on your computer. A healthy practice to keep the operating system in use safe is to daily scan the system before using it. That way you are removing the malwares before the use of the operating system and that way the chance that sensitive data is collected while it is in use is reduced.

## Bibliography

[Cobb](https://www.techtarget.com/contributor/Michael-Cobb), M. (n.d.). *Best practices for audit, log review for IT security investigations*. Retrieved from https://www.computerweekly.com/tip/Best-practices-for-audit-log-review-for-IT-security-investigations

(n.d.). *About MITRE*. Retrieved from https://www.mitre.org/

(n.d.). *Common Vulnerability Scoring System v3.0: User Guide.* Retrieved from <https://www.first.org/cvss/user-guide>

(n.d.). *Common Vulnerability Scoring System v3.0: Specification* *Document.* Retrieved from <https://www.first.org/cvss/specification-document>

(n.d.). *General Information.* Retrieved from https://nvd.nist.gov/general

Chao, G., Khau, B. (2013) *Lecture 17: Introduction to Security*

Retrieved from: http://web.cs.ucla.edu/classes/winter13/cs111/scribe/17b/

Endignoux, G., & Vizár, D. (2017). Linking Online Misuse-Resistant Authenticated Encryption and Blockwise Attack Models. *IACR Transactions on Symmetric Cryptology*, *2016*(2), 125-144.

Kamarei, M., Patooghy, A., & Fazeli, M. (2016). Unauthenticated event detection in wireless sensor networks using sensors co-coverage. *The ISC International Journal of Information Security*, *8*(1), 61-71.

[Rouse](https://www.techtarget.com/contributor/Margaret-Rouse), M. (2007, June). *Active Attack.* Retrieved from: <https://whatis.techtarget.com/definition/active-attack>

Rouse, M. ([2007](file:///C:\Users\alexa\OneDrive\Documents\School\June,%202007), June). *Masquerade.* Retrieved from: <https://searchsecurity.techtarget.com/definition/masquerade>