# Introduction to threats

## What is a threat

A threat is considered everything capable of causing harm to a system or service. This can be the disclosure or modification of data, denial of service… Any action or event that deviates from the planned function of a system. Threats come in different shapes and sizes, it can go from a small malfunction of a program to an incident where the client database of a large organization is leaked. Because the amount of types of threats is so vast, a simple classification is needed. The most simple classification is the one based on the origin of the threat.

Threats can have multiple origins:

* Deliberate: this is the most known origin of a possible threats. These threats are actively trying to exploit a certain vulnerability in a system to disrupt its service. At any given moment in time there are thousands of attacks ongoing worldwide. These attacks can vary from viruses written to disrupt a system, organized criminals that try to make a profit by stealing and selling data, spies that try to steal data from other companies or even other governments.
* Accidental: not every threat is the work of a criminal organization, a threat can also be an accident. These kind of threats can be caused by a malfunction of the system, misconfiguration of a system, power loss, accidental deletion and so on.
  + Environmental threats (floods, earthquake, lightning, tornado…) are also part of the accidental threats.

It is the goal of a cybersecurity team to counteract as much of these threats as possible. Accidental threats are most of the time counteract using hardware resources. For example the build of a flood wall around a datacenter, the installation of an UPS (Uninterruptable power source) to prevent accidental power loss, reinforcements against earthquakes… These countermeasures and type of threats go beyond the scope of this thesis so will no longer be considered.

The focus of this thesis will be at minimizing the deliberate threats. These threats are (most of the time) counteract using software resources like firewalls, anti-virus, intrusion detection systems which will be discussed later on.

## Threat management

The cybersecurity team is constantly trying to improve the defenses of an organization. This process is the most effective knowing what threats are imminent. Otherwise it would be like fighting an unknown enemy. The process of constantly analyzing, researching, quantifying possible threats is called threat management. This process can have an active (actively researching possible ) or passive (analyzing attacks after they occurred) character.

### Passive threat management: incident research

Passive threat management only kicks in when an incident or warning occurred on existent security infrastructure. This can be firewalls, intrusion detection systems… When an incident occurred the cybersecurity team will assess and review the situation to prevent the incident from reoccurring.

### Active threat management: threat hunting

Actively searching, detect and isolate possible threats that evade current security solutions. This is also called an iterative process because the pool of possible threats is constantly change so the this process has to be executed continuously in order to be effective.

During threat hunting vast amount of network data must be analyzed. It would be quite impossible to analyze all this data manual so to automate a part of the analysis, threat hunting software was developed. This software specializes in behavioral analytics because this could also zero-day threats.

* DomainTools: investigation DNS related cybercrime
* Exabeam, sqrrl: User and Entity Behavior Analytics (UEBA)

Threat hunting seems to be very effective according to an investigation by the SANS institute. They reported (in 2016): 74% reduce in attack surface, 59% faster and more accurate threat responses, 52% more threats were discovered.

## Threat models

In order to get a clear overview of the possible threats a threat model is used. This is a process where an organization can identify the possible threats. During this process the organization (the party that tries to improve their defenses) take a look at their most valuable assets from the attackers point of view. This is the fastest way to find any possible loopholes in the organization defenses. Every threat model is different but they all based on the same principles. The basic threat model consists of four steps:

* Define objectives and scopes: without a set of business objectives, security criteria the threat model will lack foundation and is likely to be ineffective.
* Decompose the system: a set of possible targets.
* Identify threats: enumeration of the possible threats to each and every possible target.
* Prioritize the threats: possible threats are sorted from most harmful/likely to the most harmless/unlikely.

### STRIDE

STRIDE is the successor of the DREAD threat model (DREAD is classified as outdated and will not be further discussed for this reason). This threat model divides threats into six categories.

* **Spoofing of user identity:** a process where a person or program presents itself as another. This can be a malicious program hiding or acting as a legitimately program (Trojan), an attackers pretending to be someone else (social-engineering), spoofing of IP-address.
* **Tampering:** these kind of threats specialize in altering data in such a way to make it harmful to a system or the users of a system.
* **Repudiation:** a repudiation attack mainly targets the logging mechanism of a system. This means that the attacker has altered the system in a way that his malicious activities won’t (or in an incorrect way) be logged. After a successful repudiation attack none of the attacks can be linked with a certain attacker afterwards. Which makes counteracting the “hidden” attack and proving criminal activity next to impossible.
* **Information disclosure:** this threat involves the exposure of information to individuals who are not supposed to have access to it. This can be: intercepted network traffic (man in the middle), changing reading permission on files. This type of threats is commonly used by whistleblowers to reveal unethical, malicious, fraudulent activities of a certain organization or government.
* **Denial of service:** one of the most popular, easiest and effective threats. This threats does not affect data in any way but it will deny any access to it. This is widely used in attacks against web servers to prevent anyone accessing a certain website or web service. This is threat is popular because is quite difficult to defend against and even large organization can become victim. GitHub, DynDNS, SpamHouse, BBC have all been a victim of DDOS attacks.
* **Elevation of privilege:** in this type of threat an unprivileged user gains privileged access to data or systems. With this privileged access the attacker is capable of compromising or destroying the entire system. Most of the previous threats lead up to this attack (all defenses have been penetrated and the attackers becomes part of the trusted system itself). A few examples are: jail breaking Apple products, rooting Android products, dirty cow attack…

### PASTA

The next threat model is fairly new but gaining popularity fast. The reason it is gaining this much popularity in such short amount of time is that it not only takes business objectives and technical requirements in to consideration but also business impact analysis and compliance requirements. This gives the modal a more dynamic approach.

Following are the seven steps in threat modeling according to the pasta model.

* **Define business and security objectives:** Capture requirements for the analysis and management of web based risks
* **Define technical scope:** Defining the scope of technical assets/ components for which threat enumeration will ensue
* **Application decomposition:** Identify the application controls that protect high risk web transactions sought by adversaries
* **Threat analysis:** Identifying and extracting threat information from sources of intelligence to learn about threat-attack scenarios used by web focused attack agents
* **Vulnerability and weakness analysis:** Analyzing the weaknesses and vulnerabilities of web application security controls
* **Attack modeling:** Attacks/Exploits Enumeration & Modeling
* **Risk & impact analysis:** Impact Analysis, Residual Risk, and Countermeasure Development

### TRIKE

This threat model shows some similarities with the STRIKE model but the big difference is that it uses a risk based approach. This model show great potential but is considered to be in the experimental phase because of its difficulty to implement and poor documentation. But since TRIKE is an open source project it is still undergoing some big changes and is starting to show some big potential. This model contains three distinct models: implementation, threat and risk.

Another big difference with other models is that TRIKE puts a great emphasis on communication between the organization and the stakeholders. The communication factor is something other models seem to neglect.

# Detecting threats

Before a certain threat can be blocked it must be identified as a threat. There are several techniques for identifying threats. These principles are used by most security infrastructure (firewall, IPS, antivirus)

## Reputation-based

The most basic filtering is done reputational. Reputation-based filtering uses lists linking certain payloads or IP addresses to a reputation index. This is a representation of how trustworthy a payload (any data that is exchanged: code, files, e-mail…) or IP address is.

These scores can be determined locally or globally.

### Local reputation

When a firewall successfully blocked an attack the source of the attack, being a file, ip-address… will get a “bad reputation”. These reputations are stored in lists locally on the firewall. If a particular source with a bad reputation tries to connect in the future, the firewall will be extra careful handling this source. Every time the firewall detects malicious activity will the source get an even worse score until the point where the firewall immediately blocks any further connections or interactions with that source.

### Global reputation

The lists of local indexes can be shared to other firewalls over the network (other firewalls in the enterprise or ideally to all firewalls over the world). This makes that a source linked to malicious activity on firewall one will already get a bad reputation on another firewall on the other side of the world. The malicious source will thus be blocked on the firewall even before it ever connected to that firewall.

This is especially useful when blocking ip-addresses. Addresses linked to botnets, tor nodes, known spammers and hackers, anonymous proxies are public available so firewalls can be extra careful handling traffic that originates from these sources. IP addresses can also be linked to a certain country with a certain reputation. For example IP addresses from North-Korea are more suspicious than an IP address from France.

## Signature-based

During signature-based detection hashes of known malicious payloads are used to determine if a given payload is malicious. This offers a quick analysis of large files (the hashes have a fixed length independent of the file size), but is easily evaded using obfuscation or polymorphic code and doesn’t protect against zero-day vulnerabilities.

At this point of time signature-based detection is the most popular especially among antivirus software. These programs use immense databases containing signatures of malicious payloads. This technique is so popular because of its simplicity. On the downside this is also the most easiest technique to evade because even the smallest change in a the code of a malware file will wake it undetectable by this technique.

## User behavior-based

User behavior analytics is the new big thing in the cybersecurity world. This technique will analyze human behavior and try to detect possible anomalies in this behavior. An anomaly is any behavior that deviates from the normal, expected behavior. This can be anything from excess requests, larger payloads, unusual connection origin, strange input… User based analytics is so versatile it can even detect threats that look perfectly fine for any other device in the security infrastructure.

For example: Bob works in a local construction firm, he’s in charge of paying all the suppliers and has full access to the firms bank account. Bob normally makes five transactions with an average value of €500 per day. This is considered Bobs normal/expected behavior. Something would be not right if Bob suddenly makes a transaction with a value of €50.000. At this moment the user behavior algorithms kick in, the transactions doesn’t need to be blocked immediately (because it could still be possible this is a legitimate transaction for a really big order) but the management will be notified immediately notified of this suspicious behavior. As it turns out Bob was about to leave the firm and tried to steal large amounts of money from the company. This kind of thread could only be detected by user behavior analytics because a firewall or intrusion prevention system doesn’t make any difference between a transaction worth €50 or 50.000.

## Heuristic-based (Sandboxing)

When a payload isn’t linked to a certain reputation index or signature it is a total guess if the payload is malicious. The only way to find out if the payload is malicious is to execute it and analyze the effects on the system executing the payload. This is called sandboxing because the possible malicious file is executed in a test environment so any adversary effect will not have any impact on critical systems.

It can take a long time to complete all of the necessary test but the results will be definitive. Because the test take a lot of time and resources a deliberate decision has to be made which files need to be sandboxed. Typically an .exe file carries more risk than an image file.

# Counteracting threats

Organizations will try to defend themselves against threats using the above mentioned techniques. These techniques will be implemented by cybersecurity infrastructure also sometimes referred to as the cyber defenses. These devices are continuously further developed to keep up with the ever changing pool of threats. In this chapter the four most used devices will be discussed, it is important to keep in mind that every device is specialized in counteracting a certain type of threat. The following is not a comparison rather an enumeration of the possible devices that can be used to reinforce the organizations defenses.

## Firewall

For counteracting the most basic and the more general threats a firewall is used. This is a device that filters the incoming and outgoing network traffic. Firewalls are implemented on borders of the network, this can be between internal networks but also between the internal network and the external network (internet). These devices categorize passing traffic using a set of rules, these rules are configured by the network administrator. The strength of the firewall depends on the strength of the rules so it is the responsibility of the network administrator to make the firewall will function correctly. Throughout the years there were several generations of firewalls. They started out quite simple but every time a new technology was developed the firewalls had to change to maintain the level of security they provided before.

### First generation: packet filters

In this quite basic system packets were filtered individually (no difference if the packet would be part of a stream or not). Every packet was filtered based on IP-address, protocol and port number. The network administrator could specific which protocols were allowed using rules. If the network administrator decided to disallow SSH traffic, the firewall would block all traffic on port 21.

This generation became somewhat obsolete once other ports than the well-known ports (0 – 1023) were used and when threats over trusted protocols started appearing.

### Second generation: Stateful filters

The following generation also worked on the fourth layer of the OSI model (transport). Making it cable of filtering traffic based on the state of the network connection. This means that the first packet of stream will be checked more thorough once this packed is considered safe the rest of the connection stream will be inspected far less. This means connections could be inspect much quicker than before.

### Third generation: Application layers

This generation of firewalls inspect packets up to the application layer, this means that from now on traffic could be filtered based on the content. This was necessary because threads were appearing over normally trusted protocols (HTTP). These firewalls are capable of only filtering out the packets with malicious content so the protocol is still available for legitimate use.

Network connections to malicious websites, transfer of virus infected files could now be filtered out by the firewall. Until the birth of TLS/SSL, these encryption protocol were designed to encrypt network traffic in order to protect it from being monitored by attackers. This is one of the best security measurements but it has one big flaw. Attackers are not able to interpret network traffic but so are firewalls. Because the content of the packets is encrypted firewalls are no longer able to filter packets based on their content. Firewalls needed a new feature and this is where “Next-Generation Firewalls” come in. This type of firewall is able to decrypt network traffic to determine whether it is compliant to the rules. This was the latest big development in firewall technology.

## Intrusion detection system

This devices specializes in detecting possible intrusions. They mostly use signature and anomaly-based detection techniques. When a possible intrusion is detected the administrator will be notified (note that this system can’t undertake any action in preventing the intrusion it can only detect the intrusion). It is usually used to either strengthen the defenses in the further or gather evidence against a certain attacker. This system is a complete passive system that just monitors the traffic and reports back on possible threats.

## Intrusion prevention system

An IPS is the active variant of an IDS. This typically sits right behind the firewall and can undertake certain actions to prevent intrusion. When an intrusion is detected there are several actions an IPS can undertake such as: reset, drop, block…

The IDS/IPS are not perfect they can still be evaded in several ways:

* Fragmentation: by splitting the attack in an excessive amount of packets so that the IDS/IPS cannot reassemble the stream in time to detect the underlying attack.
* Encrypting and tunneling: an attacker can encrypt the packets and send them over a secure tunnel so the IDS/IPS cannot interpret the content of the packets.
* Insertion: the attacker insert meaningless data in the stream that don’t change the payload meaning.

## Web Application Firewall

When all previous defenses have failed stopping the attack there is the last line of defense, the web application firewall.

### Description

A web application firewall is an application, server plug-in or cloud based service that filters, monitors, blocks HTTP traffic to and from a web application. The WAF works at the application layer to prevent application layer attacks (SQL-Injection, XSS, session hijacking…), whereas the IDS/IPS works at layer 3 of the OSI model (mostly counteract network breaching attacks).

### History

Web application firewalls were developed in the nineties in order to fill in the loopholes in the security measurements that were available at that time. These firewalls were only able to prevent attacks that used uncommon protocols/ports. Application layer attacks are mostly performed over the HTTP protocol. The problem with a stateless firewall is that this kind of firewall doesn’t make any difference between an HTTP packet which will result in a SQL-injection or a packet that makes a legit request to the database. This is where the web application firewall comes in.

When the PCI Security Standard Protocol was formed and published, the PCI DSS (Payment Card Industry Data Security Standard) web application firewalls really gained attention because this standard mandates the use of a web application firewall (or an extensive code review).

### Working

The web application firewall works like an advanced IPS. The big difference between those two is that a WAF typically sits right in front of an web application (as a proxy). This means that it just has to analyze the traffic destined for that web application. A normal IPS has to analyze all of the network traffic so it has only time to do a basic analysis. Because a WAF only has to protect a certain application it can really specialize in the threats facing that type of application.

### Implementation

A web application firewall will always be deployed right in front of the application it has to defend (webserver, database) but there is still the choice between an hardware or software implementation.

#### Software

Implementing a software based web application filter is the most preferred option for small businesses. Software products are easier to install, update and maintain. It is also cheaper most of the time but it can lack some performance (depends on host performance).

There are even some open source web application firewalls available. Following is a comparison of the five most popular open source WAFs.

##### ModSecurity

This is one of the most popular web application firewalls. It filters (allow/block traffic), monitors and logs based on “Sec Rules” (these rules are based on the core rule set of OWASP) in real-time. The engine can function as reverse proxy (a separate security layer, between external connection and web server, can be seen as a new point of failure) or can be embedded (apache module, best option to implement in already existing architecture) within the server itself. ModSecurity uses several techniques (virtual patching, ip reputation, URL encoding) to prevent application layer attacks like XSS, Trojans, information leakage…

##### IronBee

IronBee is a security framework to build your own web application firewall. This is not a finished product rather a framework to create your own tailored web application firewall. The main goals are to be minimalistic, provide simple API, modularity and allow interaction with external systems. Rules are written in LUA and come in three different types: basic matching rules, stream matching rules and external rules.

##### NAXSI

This is a quite basic but very effective working web application firewall, it specializes in counteracting XSS en SQL-Injection and it only filters PUT and GET packets. The firewall uses the implicit deny all policy, so adding rules will add exceptions that are accepted. NAXSI is only available for nginx (NAXSI stands for Nginx against XSS and SQL injection).

It distinguishes itself from the other firewalls by not relying on signatures but rather the behavior of the payload. It also comes with its built in tool to generate your own custom rules.

##### WebKnight

WebKnight is designed for Microsoft IIS, this web application firewall also provides protection against brute-force attack (this is the only web application firewall in this list that provides any protection against this kind of attack).

##### Shadow Daemon

The last firewall aims to be the most easiest of use. It still offers protection against the common attacks like the above firewalls but also against backdoor access.

#### Hardware

It’s also possible to implement the web application firewall as an appliance (hardware component). This is the preferred option for larger organizations because appliance products tend to have a better performance an can handle larger amounts of traffic. The performance of an appliance WAF is measured in throughput (volume of data that can pass through the device, this is important because all the traffic must go through the WAF before reaching the requested web service, this is the network performance) and transactions per second (Amount of HTTP and SSL transaction that can be handled, this is the processing performance).

##### Barracuda

Probably the best known web application firewall appliance is the one from Barracuda. At this moment they sell five different models. Ranging from 25Mpbs to 4 Gpbs throughput, up to 50000 SSL transactions with the price ranging from €6000 to €45000. They support HTTP(S), FTP(S), XML, IPv4 and IPv6. Barracuda assures protection against SQL-Injection, XSS, Cookie or form tampering and DDoS.

The barracuda WAFs also have some features to prevent some attacks: website cloaking (Strips identifying banners and version numbers from web server software to defeat server fingerprinting attacks), data theft protection(deep inspects all server responses to prevent leakage of sensitive information using provided default patterns) and much more.

##### Netscaler

Another popular appliance is the Netscaler from Citrix. They sell model that have a throughput up to 44Gbps and can process 60000 SSL transactions per seconds. The big difference with previous mentioned appliances is that the Netscaler are modular (hence the name). The base models can be upgraded to even further improve their performance.

##### SecureSphere

The last appliance that will be discussed is the one from Imperva, the SecureSphere. They share about the same performance as the web application firewalls from barracuda but are more oriented towards cloud integration.

#### Cloud

If the appliance performance is needed but installing one isn’t an option (cost, experience…) a cloud based solution can be interesting. Web traffic is diverted to specialized services where the traffic is analyzed before being redirected to the actual web application. An extra advantage is that only legitimate traffic is forward to the application which means that the web application won’t be flooded with malicious traffic (which is especially useful when the network performance of the application is limited). The three most popular services are Incapsula, CloudFlare, SUCURI. They will not be individually discussed because they all offer about the same features (DDoS, SQL-Injection, XSS, SPAM, login protection).

# TMP: Knowing your enemy

Fighting an unknown enemy can be though so it can be interesting at today most imminent threats. The focus will be on the web application attack because this is the main focus of this thesis.

## OWASP

The Open Web Application Security Group is an online (non-profit) organization that enables other organizations to conceive, develop and maintain applications that can be trusted. Everyone can post articles about newly discovered attacks or ways to mitigate them. OWASP is focused on web applications and is best known for their top ten Application Security Risks. This top ten is made up by data submitted by organizations word-wide. The most recent one was released in 2013 but later in 2017 a new top ten will be released.

### Top Ten

1. Injection: in this attack occurs when untrusted data (user input) is used as part of a command or query. The attacker can use this to execute undesirable commands that can tamper, destroy or reveal data. It is number one because the attack is easy to execute and can have devastating results.
2. Broken authentication and session management: Incorrect implementation of functions related to authentication and session management. This can allow attackers to compromise passwords, keys or session tokens. (authentication tokens or session id in URL)
3. Cross-site scripting (XSS): this flaw allows attackers to execute scripts in the victim browser which can lead to defaced websites, unwanted redirection… This attack shares some characteristics with injection attacks the big difference is that XSS is client side and injection attacks are server side.
4. **Insecure direct object references**
5. Security misconfiguration: a lot of application can be very secured but are misconfigured or outdated which makes them vulnerable. This can be because the developers are misinformed, or the security measurements make the application inconvenient.
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

https://www.owasp.org/index.php/Top\_10\_2013-Top\_10

# References

|  |  |
| --- | --- |
| [1] | S. Simeonova, "Threat Modeling in the Enterprise, Part 2: Understanding the Process," securityintelligence, [Online]. Available: https://securityintelligence.com/threat-modeling-in-the-enterprise-part-2-understanding-the-process/. [Accessed 15 February 2017]. |
| [2] | D. Shinder, "The Pros and Cons of Behavioral Based, Signature Based and Whitelist Based Security," techgenix, 13 November 2008. [Online]. Available: http://techgenix.com/pros-cons-behavioral-signature-whitelist-security/. [Accessed 08 February 2017]. |
| [3] | M. Rouse, "Web application firewall (WAF)," techtarget, [Online]. Available: http://searchsecurity.techtarget.com/definition/Web-application-firewall-WAF. [Accessed 06 February 2017]. |
| [4] | M. Rouse, "PCI DSS (Payment Card Industry Data Security Standard)," techtarget, [Online]. Available: http://searchfinancialsecurity.techtarget.com/definition/PCI-DSS-Payment-Card-Industry-Data-Security-Standard. [Accessed 06 February 2017]. |
| [5] | A. Ndegwa, "What is a Web Application Firewall?," maxcdn.com, 31 May 2016. [Online]. Available: https://www.maxcdn.com/one/visual-glossary/web-application-firewall/. [Accessed 20 February 2017]. |
| [6] | J. McMillan, "IDFAQ: What is the Difference Between an IPS and a Web Application Firewall?," sans.org, November 2009. [Online]. Available: https://www.sans.org/security-resources/idfaq/what-is-the-difference-between-an-ips-and-a-web-application-firewall/1/25. [Accessed 20 February 2017]. |
| [7] | C. Kumar, "5 Open Source Web Application Firewall for Better Security," geekflare, 4 September 2016. [Online]. Available: https://geekflare.com/free-transactional-hosted-marketing-email/. [Accessed 07 Februari 2017]. |
| [8] | B. Causey, "Introduction to Web application firewalls in the enterprise," techtarget, [Online]. Available: http://searchsecurity.techtarget.com/feature/Introduction-to-Web-application-firewalls-in-the-enterprise. [Accessed 06 February 2017]. |
| [9] | paloaltonetworks, "What is an intrusion prevention system?," paloaltonetworks, 2017. [Online]. Available: https://www.paloaltonetworks.com/documentation/glossary/what-is-an-intrusion-prevention-system-ips. [Accessed 20 February 2017]. |
| [10] | Trustwave, "What Can ModSecurity Do?," Trustwave, [Online]. Available: https://www.modsecurity.org/about.html. [Accessed 07 Februari 2017]. |
| [11] | OWASP, "Web Application Firewall," OWASP, [Online]. Available: https://www.owasp.org/index.php/Web\_Application\_Firewall. [Accessed 06 February 2017]. |
| [12] | Wikipedia, "User behavior analytics," Wikipedia, 08 January 2017. [Online]. Available: https://en.wikipedia.org/wiki/User\_behavior\_analytics. [Accessed 07 February 2017]. |
| [13] | owasp, "Threat Risk Modeling," owasp, 23 January 2017. [Online]. Available: https://www.owasp.org/index.php/Threat\_Risk\_Modeling. [Accessed 14 February 2017]. |
| [14] | Wikipedia, "Threat model," Wikipedia, 21 January 2017. [Online]. Available: https://en.wikipedia.org/wiki/Threat\_model. [Accessed 15 February 2017]. |
| [15] | Wikipedia, "Threat (computer)," Wikipedia, 06 January 2017. [Online]. Available: https://en.wikipedia.org/wiki/Threat\_(computer). [Accessed 14 February 2017]. |
| [16] | microsoft, "The STRIDE Threat Model," microsoft, [Online]. Available: https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx. [Accessed 14 February 2017]. |
| [17] | Wikipedia, "STRIDE (security)," Wikipedia, 29 January 2017. [Online]. Available: https://en.wikipedia.org/wiki/STRIDE\_(security). [Accessed 15 February 2017]. |
| [18] | Wikipedia, "Stateful firewall," Wikipedia, 3 December 2016. [Online]. Available: https://en.wikipedia.org/wiki/Stateful\_firewallStateful firewall. [Accessed 20 February 2017]. |
| [19] | Wikipedia, "Spoofing attack," Wikipedia, 19 December 2016. [Online]. Available: https://en.wikipedia.org/wiki/Spoofing\_attack. [Accessed 14 February 2017]. |
| [20] | applicure.com, "Software WAF vs. Appliance WAF," applicure.com, [Online]. Available: http://www.applicure.com/solutions/software-waf-vs-appliance-waf. [Accessed 20 February 2017]. |
| [21] | scmagazine, "Signature-Based or Anomaly-Based Intrusion Detection: The Practice and Pitfalls," scmagazine, [Online]. Available: https://www.scmagazine.com/signature-based-or-anomaly-based-intrusion-detection-the-practice-and-pitfalls/article/548733/. [Accessed 07 February 2017]. |
| [22] | cerberhost, "Reputation based Firewalling," cerberhost, [Online]. Available: http://www.cerberhost.com/avada\_portfolio/ip\_reputation\_based\_firewall/. [Accessed 06 February 2017]. |
| [23] | owasp, "Repudiation Attack," owasp, 19 December 2013. [Online]. Available: https://www.owasp.org/index.php/Repudiation\_Attack. [Accessed 14 February 2017]. |
| [24] | Owasp, "Real World Threat Modeling," 2012. [Online]. Available: https://www.owasp.org/images/a/aa/AppSecEU2012\_PASTA.pdf. [Accessed 14 February 2017]. |
| [25] | Wikipedia, "Privilege escalation," Wikipedia, 10 November 2016. [Online]. Available: https://en.wikipedia.org/wiki/Privilege\_escalation. [Accessed 15 February 2017]. |
| [26] | Wikipedia, "Next-Generation Firewall," Wikipedia, 9 February 2017. [Online]. Available: https://en.wikipedia.org/wiki/Next-Generation\_Firewall. [Accessed 20 February 2017]. |
| [27] | naxsi, "naxsi," naxsi, 02 August 2016. [Online]. Available: https://github.com/nbs-system/naxsi. [Accessed 07 Februari 2017]. |
| [28] | Qualis, "IronBee Reference Manual," Qualis, 04 January 2016. [Online]. Available: https://www.ironbee.com/docs/manual/ironbee-reference-manual.html. [Accessed 07 Februari 2017]. |
| [29] | citrix, "IP Reputation," citrix, [Online]. Available: https://docs.citrix.com/en-us/netscaler/11/security/reputation/ip-reputation.html. [Accessed 07 February 2017]. |
| [30] | Wikipedia, "Intrusion detection system," Wikipedia, 13 February 2017. [Online]. Available: https://en.wikipedia.org/wiki/Intrusion\_detection\_system. [Accessed 20 February 2017]. |
| [31] | Zecure, "Introduction," Zecure, [Online]. Available: https://shadowd.zecure.org/overview/introduction/. [Accessed 07 Februari 2017]. |
| [32] | Wikipedia, "General Data Protection Regulation," Wikipedia, 24 January 2017. [Online]. Available: https://en.wikipedia.org/wiki/General\_Data\_Protection\_Regulation. [Accessed 08 February 2017]. |
| [33] | EY, "EY," EY, [Online]. Available: http://www.ey.com/be/en/home. [Accessed 07 February 2017]. |
| [34] | Wikipedia, "Ernst & Young," Wikipedia, 07 February 2017. [Online]. Available: https://en.wikipedia.org/wiki/Ernst\_%26\_Young. [Accessed 07 February 2017]. |
| [35] | DenyAll, "Discover DenyAll IP Reputation Services," DenyAll, [Online]. Available: https://www.denyall.com/products/denyall-ip-reputation-service/. [Accessed 07 February 2017]. |
| [36] | dirtycow, "dirtycow," dirtycow, [Online]. Available: https://dirtycow.ninja/. [Accessed 15 February 2017]. |
| [37] | barracuda, "Configuring IP Reputation Filter," barracuda, [Online]. Available: https://campus.barracuda.com/product/webapplicationfirewall/article/WAF/ConfigIPReputationFilter/. [Accessed 06 February 2017]. |
| [38] | threatmodeler, "Comparison of Threat Modeling Methodologies," threatmodeler, [Online]. Available: http://threatmodeler.com/comparison-threat-modeling-methodologies/. [Accessed 14 February 2017]. |
| [39] | Wikipedia, "Big Four accounting firms," Wikipedia, 06 February 2017. [Online]. Available: https://en.wikipedia.org/wiki/Big\_Four\_accounting\_firms. [Accessed 07 February 2017]. |
| [40] | AQTRONiX , "AQTRONiX WebKnight - Open Source Web Application Firewall (WAF) for IIS," AQTRONiX , [Online]. Available: https://www.aqtronix.com/?PageID=99. [Accessed 07 Februari 2017]. |
| [41] | isaca, "APPLICATION THREAT MODELING," isaca, November 2013. [Online]. Available: http://www.isaca.org/chapters5/Ireland/Documents/2013%20Presentations/PASTA%20Methodology%20Appendix%20-%20November%202013.pdf. [Accessed 14 February 2017]. |
| [42] | Wikipedia, "Application firewall," Wikipedia, 29 January 2017. [Online]. Available: https://en.wikipedia.org/wiki/Application\_firewall. [Accessed 06 February 2017]. |
| [43] | M. Fainshtein, "Barracuda Web Application Firewall," Barracuda, [Online]. Available: https://www.barracuda.com/assets/docs/Datasheets/Barracuda\_Web\_Application\_Firewall\_DS\_US.pdf. [Accessed 21 February 2017]. |
| [44] | Barracuda, "Barracuda WAF," Barracuda, [Online]. Available: https://campus.barracuda.com/product/webapplicationfirewall/article/WAF/. [Accessed 21 February 2017]. |
| [45] | citrix, "Citrix NetScaler Application Firewall," citrix, [Online]. Available: https://www.citrix.com/content/dam/citrix/en\_us/documents/data-sheet/citrix-netscaler-application-firewall-datasheet.pdf. [Accessed 21 February 2017]. |
| [46] | Imperva, "Imperva SecureSphere Appliances," Imperva, [Online]. Available: https://www.imperva.com/docs/DS\_SecureSphere\_Appliances.pdf. [Accessed 21 February 2017]. |
| [47] | geekflare, "Top 3 Cloud Web Application Firewall to Stop Website Attacks (for Small to Medium Business)," geekflare, [Online]. Available: https://geekflare.com/cloud-waf-to-stop-website-attacks/. [Accessed 21 February 2017]. |