Improving Network, Data and Application Security for SMEs

Christos Tselios
Wireless Communications Laboratory
University of Patras
Achaia, Greece
ctselios@upatras.gr

Ilias Politis
Systems Security Laboratory
University of Piraeus
Piraeus, Greece
ipolitis@ssl.ds.unipi.gr

Christos Xenakis
Systems Security Laboratory
University of Piraeus
Piraeus, Greece
xenakis@unipi.gr

ABSTRACT

The evolution of Information and Communications Technology and Cloud Computing, combined with the advent of novel telecommunication frameworks such as 5G, have introduced the notion of ubiquitous connectivity combined with a seemingly vast pool of resources, storage and services. This immense transformation introduced new types of security threats mostly due to the significant increase of the attack surface, which can now be compromised by malicious users. Despite the fact that malicious attacks constantly become more and more sophisticated, SMEs and public administrations remain reluctant to invest in cybersecurity since they operate on a limited budget and are mostly focused in time to market and cost minimization. The purpose of this book chapter is to provide an overview on how the most common network-related cybersecurity attacks are orchestrated, which are the systems and services they affect the most as well as present specific design principles and guidelines for crafting platforms and frameworks capable of mitigating such attacks and ensure a certain level of secure operation.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Security, Network, Data, SME

ACM Reference Format:
Christos Tselios, Ilias Politis, and Christos Xenakis. 2022. Improving Network, Data and Application Security for SMEs. In *The 17th International Conference on Availability, Reliability and Security (ARES 2022), August 23–26, 2022, Vienna, Austria.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3538969.3544426

1 INTRODUCTION

The evolution of Information and Communications Technology and Cloud Computing, combined with the advent of novel telecommunication frameworks such as 5G, have introduced the notion of ubiquitous connectivity combined with a seemingly vast pool of resources, storage and services. This new reality has significantly lowered both management and operational costs of companies and organizations and fueled the fast growth of new online services and products. Remote accessibility and management drastically reshaped the way companies operate today. Nearly all organizations such as big/medium/small companies, public administrators and government authorities have online presence spanning from websites with basic information to complex systems offering product ordering, billing capabilities and advanced customer services. This immense transformation introduced new types of security threats mostly due to the significant increase of the attack surface, which can now be compromised by malicious users. Large companies are heavily investing time and money to protect their systems, yet this is not the case for small/medium companies due to economic constraints, lack of resources and priority for reaching the market as fast as possible.

Delivering effective cybersecurity is a highly complex problem. Each company with its services and products has its own peculiarities and requirements forcing solutions to be tailored made for each case separately. Because of this, security solutions are expensive, take too much time to deploy, are highly complex and require experts to setup and manage. Big companies are willing to pay the price to protect them since a potential breach could have severe economical impact. On the other hand, SMEs and public administrations with limited budgets are mostly focused in time to market and cost minimization and remain reluctant to invest in cybersecurity. However, as malicious attacks constantly rise and become more and more sophisticated, cybersecurity will inevitably become a topic of major discussion. Especially with the arrival of 5G, where novel and much more perplexed types of mitigation measurements will be necessary, countermeasures may involve Machine/Deep Learning techniques [1, 8, 17], network coding [2, 25] and blockchain integration [20]. The purpose of this book chapter is to provide an overview on how the most common network-related cybersecurity attacks are orchestrated, which are the systems and services they affect the most together with the direct effect they have on compromised systems and finally to present certain guidelines for the design of platforms and frameworks capable of mitigating them and operate with a certain level of security confidence. The rest of the chapter is organized as follows: Section 2 presents contemporary cybersecurity attacks while Section 3 contains the guidelines for designing a secure and robust system. Finally, Section 4 lists the conclusions and summarizes the chapter.

ARES 2022, August 23–26, 2022, Vienna, Austria

ACM ISBN 978-1-4503-9670-7/22/08…$15.00
https://doi.org/10.1145/3538969.3544426

2 CYBERSECURITY ATTACKS CATEGORIES

2.1 Attacking over the Network

2.1.1 Distributed Denial of Service. A distributed denial-of-service (DDoS) attack — or simply DDoS attack — occurs when multiple

compromised systems are used as origin points of improper network traffic, all targeting a specific service or application server. The rapid increase of inbound traffic saturates the network, overloads the attack target and renders it incapable of accommodating legitimate user requests leading to service delay, disruption or even complete shutdown. In its most common form, DDoS attack "floods" the network with useless packages of information, exploiting the universal server regulation for mandatory reply against all incoming requests which consequently leads to outage due to lack of adequate processing capacity or network bandwidth. Access to legitimate users is eventually denied while from a business perspective this translates to measurable revenue loss for as long as critical applications are down.

From the attacker perspective, using multiple points of origin for launching an attack has several advantages. A large number of compromised nodes generates substantial amounts of traffic compared to a single server, while in the same time the attack is much more difficult to be mitigated. Identifying one malicious node and drop all ingress traffic originating from it will have not affect on the overall process whatsoever, since the attack will not stop. In addition, multiple nodes may have a much stealthier behavior over the network, making the identification significantly harder. These attacker advantages introduce new challenges for potential defence mechanisms. Merely purchasing more incoming bandwidth than the existing one will most likely won't help in the long run, since the attacker only has to simply add more attacking nodes. Computational resources scale-out is also ineffective since despite all efforts the service will crash and become unavailable for periods of time. Stopping the attack is the only effective strategy but this mandates a solid understanding of how real-world DDoS attacks are launched.

For effectively launching a DDoS attack, the attackers rely mainly on malicious software which allows them to convert legit servers into bots. The bots are then aggregated into a wider network of nodes known as botnet, used by the attackers to launch the attack. An alternative approach for obtaining control over remote servers is using automated tools that exploit design/security flaws in programs that listen for connections from third party hosts. A well known DDoS tool is Stacheldraht[1] rendering the attacker capable of connecting to a first layer of compromised systems, the Handlers, used to issue commands to a second layer of compromised systems, the Agents, which in turn facilitate the DDoS attack. Each Handler can control up to a thousand Agents, using automated routines for exploiting vulnerabilities in programs that accept remote connections. Stacheldraht uses classic DoS attack methods known as bandwidth consumption attacks, mostly relying on IP spoofing [13] and amplification like smurf attacks [7].

It is also possible that simple attacks such as SYN floods [6] to appear having a wide range of source IP addresses, thus giving the appearance of a well distributed DoS. These flood attacks do not require completion of the TCP three way handshake and attempt to exhaust the destination SYN queue or the server bandwidth. Because the source IP addresses can be trivially spoofed, an attack could come from a limited set of sources, or may even originate from a single host. In general, proper classification dictates that attacks against service availability mounted from a single host to be classified as DoS while those in which a attacker exploits many systems simultaneously to attack a remote host, be classified as DDoS ones.

2.1.2 Man In the Middle attacks. Eavesdropping is defined in real life as the act of silent conversation overhearing. In networking, an "Eavesdropping attack" occurs when an unauthorized party improperly intercepts, modifies or deletes essential information transmitted between two nodes. This type of attack, also known as sniffing or snooping attack, is considered insidious since it is difficult to know they take place to begin with. Once connected to an open network, naive or cybersecurity-unaware users tend to behave as if the communication channel is absolutely safe, thus unwittingly share sensitive information including passwords, account numbers or private messages with whoever is listening. Even in cases where a layer of security is introduced to protect information leaks, an eavesdropper may still compromise the network by exploiting weak passwords, VPN security holes, or by injecting malware or network sniffers in the network pathway that will monitor and duplicate potentially critical business information. Man-in-the-Middle is a case of active eavesdropping in which the attacker makes independent connections with the target nodes and then relays messages in between, deceiving them that they are talking directly to each other over a private connection [18]. However, in reality the entire conversation is controlled by the attacker since he is now able to intercept all relevant messages traversing between the two endpoints and inject new ones.

The impact of Eavesdropping attacks may be significant, ranging from loss of privacy and financial damage to severe cases of identity theft which are really hard to mitigate. Every business holds confidential information able to lead the organization astray if becomes public. While eavesdropping, the attackers obtain vital business ideas and conversations exchanged within the organization thus compromising its privacy. In more severe cases, attackers are able to use the leaked information for getting access to even larger amount of material which can be later sold to the highest bidder.

There are some methods for tackling eavesdropping attacks or effectively minimise the damage they may inflict into a business. Primarily, it is advised to use encryption for all message exchange amongst network points. When encryption is enforced in an end-to-end manner, even if an attacker manages to intervene between a communication, the attack would be considered successful only if intercepted data can be interpreted. By using a 256-bit, also known as military-grade encryption, the attacker may gather large datasets via eavesdropping, but the content will still be safe as deciphering is nearly impossible. An additional countermeasure against eavesdropping attacks in large corporation/business environments is network segmentation. The overall computer network is divided into logically isolated parts, each accessible by specific individuals and key personnel only. This tactic boosts network traffic decongestion, improves security and prevents information leakage. However, the most important factor for preventing an eavesdropping attack is raising awareness among all employees through cybersecurity training and constant updates. Given the fact that a chain is only as strong as its weakest link, an unaware employee may unknowingly put the whole organization at risk, thus avoiding potential malware

[1] https://packetstormsecurity.com/distributed/stachel.tgz

downloading and installation, or connect to the company's backend infrastructure through open/weak networks.

2.2 Data Security breaches

2.2.1 Sensitive Data Exposure. Applications and online services often reveal sensitive data to attackers which launch successful man-in-the-middle attacks, steal digital access keys and more often intercept clear text datasets during transmission. Lack of encryption was the most common reason of data theft, regardless if the datasets were obtained from a server, while in transit or from the user's client. Yet, even in cases where encryption is applied, brute force attacks, facilitated by contemporary Graphics Processing Units (GPUs) proved capable of successfully compromising systems relying on weak key generation and management, outdated security algorithms and protocols or weak password hashing storage techniques, regardless if applied for providing an extra layer of protection to data in transit or data at rest. A compromised system often reveals to the attacker sensitive personal information of legitimate users, such as health records, credentials or financial documents, which often require protection as defined by laws and regulations such as EU's General Data Protection Regulation (GDPR)[2] or local privacy legislation.

There are several simplistic scenarios which demonstrate improper security design leading to sensitive data exposure. For instance, an application may encrypt credit card numbers in a database using automatic database encryption, yet automatically decrypt retrieved elements from the specific database. This approach makes the application vulnerable, since it allows an SQL injection attack [16] to retrieve credit card information in clear text. Another example may come from an application server which doesn't enforce Transport Layer Security (TLS) in an end-to-end manner or supports weak encryption. A malicious user capable of monitoring network traffic (i.e. in a public, insecure wireless network), may downgrade connections from HTTPS to HTTP and steal all available session cookies. By replaying the cookies, the attacker is now able to hijack authenticated sessions and access or modify all transported data. A third example is of a password database which uses unsalted or simple hashes to store passwords. If a file upload flaw is identified, an attacker could potentially retrieve the password database and expose all unsalted hashes via a rainbow table of pre-calculated hashes. Even if the hashes were salted, those generated by simple or fast hash functions can be compromised by a GPU. These examples aim to prove that data exposure needs to be addressed from the early phases of application design and by having the broader picture in mind.

2.2.2 Spam and Phishing Attacks. Spam is defined as massively distributed, irrelevant or unsolicited messages sent over the Internet, aiming to trick recipients and collect valuable pieces of personal information. Spam content varies ranging from plain advertising to consulting services and is often considered the responsible for phishing scams, fraud, privacy threats and malware spread. Attackers using spam messages to (i) engage recipients into advance-fee scams, a form of fraud that typically involves promising the victim a significant financial benefit in return for a small up-front payment. If the victim makes the payment, the attacker constantly invents additional fees or in most cases simply vanishes (ii) create backlinks to their own website through false and irrelevant comments inserted in the victim's website. The attacker uses software, such as ScrapeBox[3], to find potential targets and overflow them with comments. The comments are useless to the victim, but create blacklinks to the spammer's website increasing its visibility and potential revenue.

Spamming is considered economically viable due to the low operating cost for the attackers, which only covers mailing list management, domain names, limited backend infrastructure and possible IP ranges purchase. In addition, attackers do not face criminal charges or held accountable for mass email submission. Main costs related to lost productivity and fraud, are imposed to both the public and the Internet service providers, which have been forced to add extra capacity to cope with the volume. Spamming has been the subject of legislation in many jurisdictions. With regards to SMEs, spam creates a communication service overload since businesses may need to pay a premium to provides or third party software vendors to efficiently filter electronic messages.

Phishing on the other hand is the fraudulent attempt to obtain sensitive information such as usernames, passwords and credit card details by disguising oneself as a trustworthy entity in an electronic communication. Typically carried out by email spoofing it often directs users to enter personal information at a fake website which matches the look and feel of the legitimate site. Phishing is an example of social engineering techniques being used to deceive users. Users are often lured by communications purporting to be from trusted parties such as social web sites, auction sites, banks, online payment processors or IT administrators. Attempts to deal with phishing incidents include legislation, user training, public awareness, and technical security measures, often paired with meticulously maintained phising website inventories [5], often even pragmatically populated [26] used for blacklisting or machine learning-based identification algorithm training.

2.3 Application-related attacks

2.3.1 Injection Attacks. Injection attacks is probably the most common and successful attack class on the internet, due to its large attack surface, type variation and countermeasure complexity. The main characteristic of this type of attack is identification and exploitation of specific flaws which allow malicious users to relay and execute code through an application to a third party node. Notable examples of injection attacks include Cross-Site Scripting (XSS), SQL Injection, Header Injection, Log Injection and Full Path Disclosure. As most web applications rely on operating system features and external programs to support their functionality, an attacker may include a variety of calls to numerous functional entities ranging from the operating system to the backend databases. In addition, it is also possible for an attacker to inject special (meta) characters or command modifiers into HTTP request information, having the web application blindly pass them on the end system for execution.

A particularly widespread and perilous form of injection attack is SQL injection. The attacker in order to exploit an SQL injection flaw, must be able to identify a parameter that the target web

[2] https://ec.europa.eu/info/law/law-topic/data-protection_en

[3] http://www.scrapebox.com/

application directly forwards to a database. Once such a parameter is identified, the attacker becomes capable of deceiving the application into forwarding improper queries to the database, by carefully embedding malicious SQL commands into the parameter content. The consequences of such attacks can be devastating, since the attacker can copy, corrupt, or destroy the content of the now compromised database. In general, there is no pattern or specific level of complexity for identifying injection vulnerabilities. Similarly, the consequences of successful injection attacks may range from having a trivial effect to complete system outage and service disruption. In any case, since contemporary applications heavily rely on external calls the likelihood of an application to contain an injection flaw should always be considered as high.

As all contemporary web applications allow external command execution such as system calls, shell commands, and SQL requests, the susceptibility of an external call to command injection is directly linked on how the call is conducted. However, it should be stated that almost all external calls can be attacked if the web application is not properly implemented. For instance, a malicious parameter could change the results of a system call which normally retrieves a file path to track another similar file path belonging to a different user, or SQL queries could be modified by adding additional 'constraints' to a where clause to gain access to or modify unauthorized data.

2.3.2 Cross Site Scripting Attack. Security on the web is based on a series of concepts which create a functional baseline that all services depend upon. One of these concepts is the *same-origin policy*, which states that if an entity is granted permission of accessing web browser resources, then content from any URL having the same (i) URI scheme, (ii) hostname and (iii) port number also inherits the same permission.

Attacks categorized as cross-site scripting (XSS) [4] ones exploit known vulnerabilities in web-based applications, their underlying hardware infrastructure or affiliated plug-in systems. The attacker is able to insert malicious content on top of the legitimate content which is delivered by the compromised node. When the combined content is inspected in the client-side upon arrival it is marked as being sent by a trusted source therefore is granted fully operational permissions. By injecting malignant scripts into web pages, attackers are rendered capable of getting elevated access privileges to resources like sensitive page content, session cookies and private user information that is maintained in the browser on behalf of the user.

There are two prime categories of XSS vulnerabilities: persistent and non-persistent with some experts further diving these groups into those caused by server-side code flaws and the Document Object Model (DOM)-based ones which take place on client-side. The non-persistent XSS vulnerability is probably the most common type and can be exploited when web client-oriented data such as HTTP query parameters, is directly utilized by the backend without any proper content sanitation. Given the fact that HTML documents are structured using a flat, serial schema containing control statements, formatting elements all fused with the actual content, it is possible that non-validated data inflicting markup injection. As mentioned in [9], an XSS flaw is possible when site-embedded search engines do not properly reject HTML control characters from the results page. However it is also common for non-persistent attacks to be delivered through innocent-looking URLs pointing to a trusted site but containing hidden web attack vectors, including XSS. By clicking the link, the victim's browser executes the injected script and the attack is instantly launched.

The persistent XSS vulnerability is a more dangerous variation of the cross-site scripting flaw: it occurs when the data injected by the attacker is permanently stored in the server and then constantly served as results in HTTP requests of all legitimate users [9]. The malicious script is automatically rendered, without the need of individually targeting victims or redirect them to third-party websites. In its most sophisticated implementation, the malicious code can be self-propagated across accounts, thus creating a type of client-side worm [9].

DOM-based XSS attacks take place in cases that web applications stores data to the Document Object Model without first enforcing any type of data sanitation process or algorithm. This exact data can be manipulated by the attackers to include XSS content such as malicious JavaScript (JS) code on the target web page. It should be stated here that DOM is a convention used to represent and automate object-related functionality in any HTML document. Consequently, all HTML documents have associated DOMs consists of object which represent document properties from the browser point of view. During the execution of a client-script, it parses the DOM of the HTML page, is accessing its properties and is in a position of changing their values. The attacker can simultaneously use a variety of DOM objects to launch an XSS attack. In general, JS frameworks, single-page applications and APIs which dynamically insert data to a web page should always be considered as the attack surface of DOM XSS.

2.3.3 Access Control. The scope of Access control is to enforce predefined policies related to user permissions. When operating normally, Access control prevents unauthorized information disclosure and data modification or discard by blocking low-privileged users. However, access control is also possible to succumb in attacks launched by malicious users. Some common attacks involve (i) bypassing access control checks by modifying the application URL, the internal application state, or through a custom API attack tool like Metasploit[4], (ii) metadata manipulation such as tampering a JSON Web Token, a cookie or a hidden field to achieve access privilege elevation or nullifying the whole validation process and lastly (iii) unauthorized API access through efficient Cross-Origin Resource Sharing (CORS) misconfiguration or missing access controls for POST, PUT and DELETE calls.

3 MITIGATING CYBERSECURITY ATTACKS

3.1 Best Practices for Network Security

The rise of mobility as integral part of contemporary technical workforce makes remote access a crucial service. However, this also increases the organization's attack surface with devastating consequences in case there is a breach. Compromising a company network allows attackers to launch a direct attack on the whole organization by obtaining privilege escalation and gain control of core components. Once inside, a backdoor or Remote Access Trojan

[4] https://www.metasploit.com/

(RAT) typically has little difficulty connecting to a Command and Control (C2) server using an outbound call to an external system, while freely available network traffic generators[5] [10, 14, 15] and DDoS tools can be configured and controlled in forming botnets capable of generating valid internet user traffic and flood targeted websites. It becomes obvious that for ensuring cybersecurity for SMEs, such issues need to be addressed in a holistic manner, applying (i) end-to-end remote access strategies, (ii) network segmentation and implementing network security zones, and (iii) intelligent load balancing and multilayer DDoS service protection.

3.1.1 Providing secure Remote Access for partners, customers and employees. Advanced remote access capabilities allow users outside the company's core network infrastructure to access applications, services and data. The existence of a dedicated node which monitors, controls and secures remote user privileges is of paramount importance. To be more specific, this dedicated node, must be capable of

- Extending remote access and Single sign-on (SSO) functionality to all applications.
- Unifying infrastructure for reducing access method proliferation.
- Intercepting incoming traffic acting as a highly capable full reverse-proxy gateway before forwarding it to the applications on the network backend.
- Providing a single URL for consolidating all third-party solutions needed to support all types of access scenarios.

A very common method of delivering seamless access capabilities is a full Secure Sockets Layer (SSL) virtual private network (VPN) configured to provide a direct network-level connection to the datacenter. However, other solutions also exist for instance NGINX Plus[6] or Citrix Application Delivery Controller (ADC)[7] which provides a specialized proxy service based on its Independent Computing Architecture (ICA) protocol for connecting privately hosted applications with the end-user equipment [22, 23]. As with SSL VPN, all data transmitted between the client and datacenter is encrypted. Such solutions are commonly delivered over a single URL which provides end users a unique entry point for remote access to web and SaaS applications from any device, with the ability to also have two-factor authentication, SSO and Federation configured. It should be stated here that such solutions should also be specifically designed to simplify and centralize access control and visibility by providing a single point of configuration and enforcement. Feature delivery should also be blocked based on client and server IP and port as well as user and group membership. Virtual channels access such as cut-and-paste, mapping, client drive mapping or printing should be enabled in a per-application manner, to provide the right level of access.

3.1.2 Network Segmentation: Implementing Network Security Zones. Network Segmentation is a strategy aiming to extend the rule of least privilege to all network infrastructure and interconnected hosts by properly implementing security zones. These zones operate specifically for minimizing user access to sensitive applications and data and are applied through firewalls and gateways. Firewalls and gateways restrict traffic to their respective zones, reducing lateral movement and attack surface to contain the blast radius of a breach. Essential network segmentation guidelines dictate that firewalls and gateways must support:

- Authentication and proxy of client connections in the necessary demilitarized zone (DMZ) - a physical or logical subnetwork that contains and exposes an organization's external-facing services, to block malformed packets and malicious requests at this point.
- Optimization, multiplexing and rate limiting of connections to backend servers to protect their resources.
- A software-defined architecture that uses virtualization to enable the hardware platform to be securely carved up into separate and unique instances, each with separate SLAs and assigned memory, SSL, CPU and virtual NICs that are either shared or dedicated.

It is important for all network nodes to be designed, implemented and deployed with segmentation in mind. More specific:

- Traffic domains need to segment traffic for different applications and tenants into fully isolated network environments on a single appliance.
- Internal administrative partitions must segment individual appliances into separate resources with dedicated administration and separate login UI, views, configuration files and logging.

This approach ensures that even in case of a successful attack, the fallout will be contained into one segment and the network infrastructure will never be compromised as a whole.

3.1.3 Improving Availability through Intelligent Load Balancing and Multilayer Denial of Service Protection. Network and service availability is challenged daily by both hardware and software failures as well as DDoS attacks that disrupt services through the exhaustion of bandwidth, compute and memory resources. This renders load balancers network nodes of paramount importance, since they must be capable of seamlessly distributing incoming client requests across multiple servers hosting web applications and content. This prevents any one server from becoming a single point of failure and, together with utilization optimization methods such as Least Connection or SNMP-based metrics, improves overall application availability and responsiveness. Global Server Load Balancing (GSLB) provides an additional layer of protection, failover and optimization for organizations with multiple sites and geographically distributed services. As part of a multilayer approach to availability, any company network must enforce mechanisms for delivering:

- DDoS protection – preferably by having a dedicated node intelligent enough to (i) monitor client connection and request parameters to prevent flood attacks such as SYN, UDP, ICMP and Smurf [7], (ii) capable of proxying the connection until a valid application request has been submitted.
- SSL/TLS offloading – by proxying, validating and, if needed, rate-limiting connections, network infrastructure should be

[5] https://trex-tgn.cisco.com/
[6] https://www.nginx.com/products/nginx/load-balancing/
[7] https://www.citrix.com/products/citrix-adc/

able to protects web services against attacks such as HeartBleed[8], and Shellshock[9] that target SSL/TLS vulnerabilities.
- Surge protection and priority queuing – load balancers must enforce mitigation policies against traffic spikes and surges that can overload backend servers involving caching and prioritizing connections, and then delivering them as the server load is reduced so that none are dropped. DNS protection is also a necessity while the support for Domain Name System Security Extensions (DNSSEC) [11] to protect against forged and corrupted host records spreading to new targets is a welcome addition to the cybersecurity arsenal.

3.2 Enhancing Data Security

Data of all kinds, including legal documents, contracts, R&D data, marketing and sales info, entertainment media or any other form of intellectual property, constitute vital organizational assets that must be protected. Over the last few years, data breaches have resulted substantial amount of compromised records, many with personally identifiable information (PII), including credit card numbers, Social Security numbers, dates of birth, driver's license, addresses, health records and government records come even containing fingerprints and security clearance data.

However, not every breach results from hacking, malware and other attacks. Other causes include unintended disclosure, payment card fraud, insider fraud, loss of documents, loss of media, and loss of both mobile and stationary devices. In addition, the popularity of consumer-grade cloud storage among users is especially problematic, moving data off the trusted network to servers outside the organization's control and the security regulations it enforces [12, 19, 21]. Thus said, it becomes important to enhance data security as a whole through techniques that promote (i) centralized monitoring and data egress control, (ii) data encryption and (iii) secure data sharing.

3.2.1 Centralization: Centralize, Monitor and Control Data Egress. In contemporary virtualized ecosystem environments, all data are aggregated in secure locations inside the organization's datacenter. The core application functionality is run on the server and interacts with the user equipment with only the necessary mouse clicks and keystrokes. This approach improves security caused by lost, stolen or destroyed endpoints. Organizations can further protect against bulk data loss by preventing file and database transfer even to authorized workstations. To prevent data from being saved on removable media such as USB drives, emailed among users, printed out or otherwise exposed to loss or theft, policies should be centrally administered for controlling users' ability to save, copy, print or otherwise move data. Some notable device policies to further enhance data security include:

- the isolation of client-side data from applications, by blocking virtual channels such as client drive mapping, print, and copy/paste.
- folder redirection allowing user folders mapping to a central file storage location in the datacenter.
- enforcing restrictions on where files can be saved to protect against loss, theft or the destruction of the endpoint.

3.2.2 Containerization: Encrypt Data both in Transit and at Rest. In natively-run mobile applications data is locally stored, and a strategy that significantly increases the the risk of data leakage and loss. Insecure mobile data storage must be addressed by enforcing containerization and encryption in all applications, devices and endpoints. More specific, through containerization (or alternatively application-level segmentation) the application data resides inside the container in which is executed and cannot be accessed by external applications. This approach is also convenient for data encryption since any isolated container within an endpoint can be easily encrypted in an independent manner, mitigating against data loss.

In addition, as the concept of bring-your-own-device is extremely popular among organizations and most importantly startups, it becomes essential to separate personal and business applications along with their associated data. Contemporary solutions available from VMWare[10], Citrix[11] and Microsoft [12], leverage industry-standard encryption for application data either at compile time or via wrapping technology. All application data is stored in a secure container that encrypts both files and embedded SQL technology on the devices. Data held in local database files is encrypted using AES-256 [3].

3.2.3 Secure Sharing: Enable Secure File Sharing to Reduce Data Loss. As employees, customers and business partners seek to collaborate efficiently, they always demand an easy way of data sharing and are constantly trying to find the path of least resistance to do so. This sometimes includes third-party solutions that are out of the visibility, approval or control of the authorized IT, leading to data sprawl and non-secure file sharing via USB drives, the Internet and personal cloud services that often lack either basic or advanced controls against data leakage. Additional solutions based on communication protocols such as FTP, lack secure authentication having credentials transmitted in cleartext. Last but not least users rely on unencrypted email, which may also have catastrophic results if files are accidentally sent to unauthorized individuals inside and outside the organization. Such a perplexed problem demands a unified solution addressing security characteristics such as (i) Authentication, (ii) Authorization, (iii) Auditing and (iv) Encryption.

- **Authentication** could be improved by implementing multiple two-factor and two-step authentication methods some of which are token-based or smartphone-dependent (SMS or push notifications to affiliated, secure applications).
- **Authorization** could be ensured by a central, IT-monitored repository which will integrate the ability to generate file links to be forwarded via e-mail but having the IT controlling, monitoring and possibly blocking the overall process. For additional data protection, an expiration date could be set to each link while the owner could also be capable of remotely wiping-out content stored in mobile devices in case of loss or theft.
- **Auditing** involves mechanisms for tracking and logging all user activity, including data access and sharing in an attempt

[8] https://heartbleed.com/
[9] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
[10] https://www.vmware.com/products/workspace-one.html
[11] https://www.citrix.com/products/citrix-endpoint-management/
[12] https://www.microsoft.com/en-ww/security/business/microsoft-endpoint-manager

to support compliance requirements and provide clarity on data utilization. This allows organizations to monitor all data-related activity and intervene once an attack or malicious activity is identified.

- **Encryption** must be implemented in a specific way: each file must be encrypted using a unique key before being copied to its permanent location and decrypted before being downloaded to the user's browser. Finally, encryption keys should not be stored in the same server as the files for ensuring that physical access to the storage server does not automatically grant access to the content of the files stored there.

3.3 Improving Application Security

As already stated, all types of applications are popular targets for exploitation. Even if security researchers track a vulnerability before hackers do, the overall remediation action for the entire organization may take months before it is finalized. Moreover, it is quite common that many successful breaches have exploited known vulnerabilities with existing patches which have never being applied into legacy applications. Especially web applications are considered even more vulnerable due to poor security configuration, incomplete patch management of the underlying operating system, vulnerabilities in the coding language, or unpatched and zero-day vulnerabilities in third party dependencies. Legacy or unsupported applications risk attacks that tamper with fields, overflow buffers or perform command injection and remote code execution. Application-layer attacks are well above the controls provided by network firewalls and intrusion detection/prevention systems (IDS/IPS), which don't understand logic attacks. Mitigating application-layer attacks requires thinking in a more abstract layer [24] and must enforce techniques related to (i) centralization of virtualized application and encrypted content delivery, (ii) containerization and (iii) meticulous inspection.

3.3.1 Centralization: Virtualize Applications and Require Encrypted Delivery. Application virtualization protects sensitive data by centralizing apps in the datacenter and allowing only a pixelated representation of the application to reach the endpoint—no actual data transfer occurs. Virtualization also allows the classification of applications based on their security requirements; sensitive apps must be siloed onto dedicated servers within a separate network segment with different sensitivity classifications and restrictions, and multiple isolated versions of web browsers should be published to address diverse security and legacy requirements of web apps. Through centralization it is possible to perform OS patches, service packs, hot fixes, and configuration updates on a single master image, greatly accelerating testing and rollout. Endpoint-based attacks such as memory or RAM scraping no longer present a risk. The spectrum of benefits this approach provides renders it suitable for all modern business environments, hence highly recommended.

3.3.2 Containerization: Manage Mobile Apps to Prevent Data Loss. Best practices for mobile application security must be also based on containerization, a form of segmentation at the device level. As already mentioned in a previous section, users should be able to use a single device with both personal and business applications, with the later being directly managed by IT. Container-based security measures must include encrypted storage, granular application data control and data wipe policies, in order to effectively extend hardware, operating systems and individual application security. For effectively augmenting application security through containerization, any contemporary solution must support the following services:

- Dedicated Micro-VPN tunnels for native mobile applications ensure that internal network resources are not exposed to ingress traffic toward personal applications which always could be infected with malware. As always, sessions must be encrypted using SSL/TLS encryption protocols.
- Device validation is of paramount importance since containerization alone can't ensure security for a device that has been jailbroken or rooted by its owner to allow the installation of pirated or non-validated applications. This is the main reason for the deployment of endpoint management solutions [23] specifically designed to support device status validation and block jailbroken devices before the initial enrollment.

3.3.3 Inspection: Protect Web Applications Against Attacks. Web applications expose a highly vulnerable attack surface with direct connectivity to databases containing sensitive customer and company information. Attackers often launch customized attacks on specific applications rendering identification by network-layer security devices such as intrusion protection systems and network firewalls nearly impossible. This approach leaves web applications exposed to application-layer attacks using known and zero-day exploits.

In addition, since web applications are often prone to DDoS attacks, protection must extend beyond the network and session layers. More specific, for mitigating a DDoS attack, the gateway must be able to block or throttle traffic that appears to be as valid at the network layer. The method involves an entity able to challenge client requests to ensure that their origin is a valid browser. Requests coming from bots and scripts typically cannot answer the challenge properly and are discarded. When POST requests are involved the process must be slightly different. The POST request is first checked for a valid cookie. Potential lack of the specific cookie must trigger an alarm and raise awareness since the platform is potentially under attack. However, the gateway should first make a request to the client demanding information resubmission using a new cookie, which becomes invalid after a predefined period of time. From that point on, every response to the client is sent using the new cookie. During an attack, all cookies sent beforehand must become invalid, while new connections as well as connections that cannot provide valid cookie data should be placed into a low-priority queue. To further ensure proper application behavior it is necessary to enforce both positive and negative security models. The positive security model understands good application behavior and treats all other traffic as malicious. In real-world scenarios, this is the only proven approach for delivering zero-day protection against unpublished exploits. Administrators should be able to create managed exceptions and relaxations when an application's intended and legal behavior might otherwise cause a violation of the default security policy. On the other hand, using a negative security

model involves a mandatory and constant scanning against known attacks using thousands of automatically updated signatures. The advanced web application protection profile must also add session-aware protections to protect dynamic elements such as cookies, form fields and session-specific URLs. Attacks that target the trust between the client and server are consequently stopped, making this type of mitigation strategy imperative for any application that processes user-specific content.

4 CONCLUSIONS AND FUTURE WORK

Delivering effective cybersecurity is a complex problem since each organization has highly differentiated requirements which dictate tailored solutions per case. This renders security solutions to be expensive, time-consuming to develop and deploy, with a high degree of complexity that often requires expert knowledge to setup and manage. However, when viewed in a layered perspective, security attacks can be partially grouped. The purpose of this book chapter was to provide an overview on how the most common network-related cybersecurity attacks are orchestrated, which are the systems and services they affect and most importantly to present certain guidelines for the design of platforms and frameworks capable of mitigating them and operate with a certain level of security confidence.

REFERENCES

[1] Vipindev Adat, Tafseer Akhtar, Ilias Politis, Christos Tselios, and Stavros Kotsopoulos. 2019. Towards Secure Network Coding Enabled Mobile Small Cells. In *2019 IEEE Global Communications Conference (GLOBECOM)*. 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9014127

[2] Vipindev Adat Vasudevan, Christos Tselios, and Ilias Politis. 2020. On Security Against Pollution Attacks in Network Coding Enabled 5G Networks. *IEEE Access* 8 (2020), 38416–38437. https://doi.org/10.1109/ACCESS.2020.2975761

[3] Joan Daemen and Vincent Rijmen. 2002. *The Design of Rijndael*. Springer-Verlag, Berlin, Heidelberg.

[4] Mohit Dayal Ambedkar, Nanhay Singh Ambedkar, and Ram Shringar Raw. 2016. A comprehensive inspection of cross site scripting attack. In *2016 International Conference on Computing, Communication and Automation (ICCCA)*. 497–502. https://doi.org/10.1109/CCAA.2016.7813770

[5] Murat Karabatak and Twana Mustafa. 2018. Performance comparison of classifiers on reduced phishing website dataset. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*. 1–5. https://doi.org/10.1109/ISDFS.2018.8355357

[6] L. Kavisankar and C. Chellappan. 2011. A mitigation model for TCP SYN flooding with IP spoofing. In *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*. 251–256. https://doi.org/10.1109/ICRTIT.2011.5972435

[7] Sanjeev Kumar. 2007. Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet. In *Second International Conference on Internet Monitoring and Protection (ICIMP 2007)*. 25–25. https://doi.org/10.1109/ICIMP.2007.42

[8] Aris S. Lalos, Athanasios P. Kalogeras, Christos Koulamas, Christos Tselios, Christos Alexakos, and Dimitrios Serpanos. 2019. *Secure and Safe IIoT Systems via Machine and Deep Learning Approaches*. Springer International Publishing, Cham, 443–470. https://doi.org/10.1007/978-3-030-25312-7_16

[9] Miao Liu, Boyu Zhang, Wenbin Chen, and Xunlai Zhang. 2019. A Survey of Exploitation and Detection Methods of XSS Vulnerabilities. *IEEE Access* 7 (2019), 182004–182016. https://doi.org/10.1109/ACCESS.2019.2960449

[10] Rufael Mekuria, Michael McGrath, Christos Tselios, Dirk Griffioen, George Tsolis, and Shahar Beiser. 2016. KPI Mapping for Virtual Infrastructure Scaling for a Realistic Video Streaming Service Deployment. In *IEEE QoMex'16 Proceedings*. IEEE Conference on Quality of Media Experience.

[11] Moritz Müller, Taejoong Chung, Alan Mislove, and Roland van Rijswijk-Deij. 2019. Rolling With Confidence: Managing the Complexity of DNSSEC Operations. *IEEE Transactions on Network and Service Management* 16, 3 (2019), 1199–1211. https://doi.org/10.1109/TNSM.2019.2916176

[12] Stavros Nousias, Christos Tselios, Dimitris Bitzas, Olivier Orfila, Samantha Jamson, Pablo Mejuto, Dimitrios Amaxilatis, Orestis Akrivopoulos, Ioannis Chatzigiannakis, Aris S. Lalos, and Konstantinos Moustakas. 2018. Managing nonuniformities and uncertainties in vehicle-oriented sensor data over next generation networks. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 272–277. https://doi.org/10.1109/PERCOMW.2018.8480342

[13] Opeyemi A. Osanaiye. 2015. Short Paper: IP spoofing detection for preventing DDoS attack in Cloud Computing. In *2015 18th International Conference on Intelligence in Next Generation Networks*. 139–141. https://doi.org/10.1109/ICIN.2015.7073820

[14] Ioannis Prevezanos, Andreas Angelou, Christos Tselios, Alexandros Stergiakis, Vassilis Tsogkas, and George Tsolis. 2017. Hammer: A real-world end-to-end network traffic simulator. In *2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 1–6. https://doi.org/10.1109/CAMAD.2017.8031618

[15] Ioannis Prevezanos, Christos Tselios, Andreas Angelou, Michael McGrath, Rufael Mekuria, Vassilis Tsogkas, and George Tsolis. 2017. Evaluating Hammer Network Traffic Simulator: System Benchmarking and Testbed Integration. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 1–6. https://doi.org/10.1109/GLOCOM.2017.8254081

[16] Amirmohammad Sadeghian, Mazdak Zamani, and Azizah Abd. Manaf. 2013. A Taxonomy of SQL Injection Detection and Prevention Techniques. In *2013 International Conference on Informatics and Creative Multimedia*. 53–56. https://doi.org/10.1109/ICICM.2013.18

[17] Parvinder Singh Saini, Sunny Behal, and Sajal Bhatia. 2020. Detection of DDoS Attacks using Machine Learning Algorithms. In *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*. 16–21. https://doi.org/10.23919/INDIACom49435.2020.9083716

[18] Ch. Tselios, K. Birkos, P. Galiotos, S. Kotsopoulos, and T. Dagiuklas. 2012. Malicious threats and novel security extensions in P2PSIP. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. 746–751. https://doi.org/10.1109/PerComW.2012.6197612

[19] C. Tselios, I. Politis, K. Birkos, T. Dagiuklas, and S. Kotsopoulos. 2013. Cloud for multimedia applications and services over heterogeneous networks ensuring QoE. In *2013 IEEE 18th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 94–98. https://doi.org/10.1109/CAMAD.2013.6708096

[20] C. Tselios, I. Politis, and S. Kotsopoulos. 2017. Enhancing SDN security for IoT-related deployments through blockchain. In *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. 303–308. https://doi.org/10.1109/NFV-SDN.2017.8169860

[21] Christos Tselios, I Politis, V Tselios, S Kotsopoulos, and T Dagiuklas. 2012. Cloud Computing: A Great Revenue Opportunity for Telecommunication Industry. In *FITCE Congress (FITCE), 51st, 6, Poznan, Poland*.

[22] Christos Tselios and George Tsolis. 2016. A Survey on Software Tools and Architectures for Deploying Multimedia-Aware Cloud Applications. In *Lecture Notes in Computer Science: Algorithmic Aspects of Cloud Computing*. Vol. 9511. Springer International Publishing, 168–180.

[23] Christos Tselios, George Tsolis, and Manos Athanatos. 2020. A Comprehensive Technical Survey of Contemporary Cybersecurity Products and Solutions. In *Computer Security*, Apostolos P. Fournaris, Manos Athanatos, Konstantinos Lampropoulos, Sotiris Ioannidis, George Hatzivasilis, Ernesto Damiani, Habtamu Abie, Silvio Ranise, Luca Verderame, Alberto Siena, and Joaquin Garcia-Alfaro (Eds.). Springer International Publishing, Cham, 3–18.

[24] Ioannis Tzolas and Christos Tselios. 2017. Virtual job management and scheduling for media services. In *2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 1–6. https://doi.org/10.1109/CAMAD.2017.8031620

[25] Vipindev Adat Vasudevan, Tafseer Akhtar, Christos Tselios, Ilias Politis, and Stavros Kotsopoulos. 2021. Study of Secure Network Coding Enabled Mobile Small Cells. In *ICC 2021 - IEEE International Conference on Communications*. 1–5. https://doi.org/10.1109/ICC42927.2021.9500614

[26] Farashazillah Yahya, Ryan Isaac W Mahibol, Chong Kim Ying, Magnus Bin Anai, Sidney Allister Frankie, Eric Ling Nin Wei, and Rio Guntur Utomo. 2021. Detection of Phising Websites using Machine Learning Approaches. In *2021 International Conference on Data Science and Its Applications (ICoDSA)*. 40–47. https://doi.org/10.1109/ICoDSA53588.2021.9617482