Operations Security Documentation

Release (latest)

Mikko Ohtamaa

October 14, 2015

Contents

Note: This is a living, work-in-progress, document. We kindly ask that you not to refer to this document or use the material presented herein until the authors release the version 1.0.

This is an online guide for operations security (OPSEC) for Internet services.

Operations security (OPSEC) is a multidisciplinary approach for protecting information and services. Though the term has a wider general meaning, this guide discusses OPSEC in the context of securing Internet services. This includes, but is not limited to, protecting information from industrial espionage, blackhat hackers, law enforcement, social engineering, and mass surveillance. Please read further a introduction to OPSEC on Wikipedia.

CHAPTER 1

About this guide

OPSEC - Operations Security Guide is a security guide and assessment tool for developing sensitive Internet services.

The aim is to provide transparent, lightweight and practical ways to develop and evaluate the security of an Internet service from a holistic viewpoint. The scope of the guide is interdisciplinary, covering the full range of business operations from team computer and mobile phone setup to safe software development practices. Read more about how and why this guide was created.

The guide presents 49 assesment points to evaluate different aspects of team and service security. The security assessment points are referred in 31 historical security incidences which could have been avoided if the operators had followed practices presented here.

CHAPTER 2

Chapters

2.1 About the author

With passion to technology and open source, Mikko Ohtamaa has worked in technology startups and mobile companies since mid nineties. Before a graduation in MSc. in Industrial Engineering and Management from University of Oulu, Mikko worked for mobile web pioneers such as Nokia, iobox and MyOrigo. Later Mikko pursued entrepreneurial career, first in software development consultancy, then as the CTO of LocalBitcoins.

Mikko's professional past includes architecture design for international medical systems where privacy and proper authorization played a key role. When working as the CTO of LocalBitcoins, a global Bitcoin exchange, he was protecting the financial assets of thousands of customers. Building one of the top high-value targets for cybercriminals, it showed what is is to be at the receiving end of phishing, denial or service, black hat and other attacks. This responsibility, where a security compromise could mean the end of business, harming millions of people, sparked to write about information security from a practical viewpoint. The security should not be an external component to business activities, but something that horizontally cuts across organization.

Mikko wrote his first bits of HTML in 1995. His favorite software development ecosystems are Python and JavaScript. You can follow his blog opensourcehacker.com or Twitter account where web development and information security-themed articles appear regularly.

2.2 Audience

Not all Internet services are equally critical. This guide is aimed for teams who themselves are developing services dealing with private and financial data, though it can be used as a reference for other kind of services. The guide is not geared towards your average company business card website, software product deployments or generic IT administration.

The audience of the guide includes, but is not limited, to

- Website developers
- System administrators and infrastructure operators
- Technology and security officers
- ...and everybody who is responsible for making sure they do not get hacked

The guide assumes the reader has the basic understanding how Internet services are developed and operated. The developers and system administrators can use the guide as a reference when building and assessing security of their organizations. Furthermore, with the evaluation points provided by the guide a third party can audit an organization and provide a public track record of the matter.

2.3 Sensitive and high value services

Sensitive services generally include

- **Hosting services**: Running and managing somebody else's software and data sets you in a privileged position to abuse or to be exploited. See *Linode*.
- **Banking and financial services**: provide access to the assets of the users.See *Bitstamp*.
- **Health services**: sensitive private information which can be used against the users. See *Linode*.
- **Dating service**: sensitive private information which can be used against the users. See *Ashley Madison*.
- **Government and legal**: having government service compromised is not good for politics.
- **Digital non-tanglible services**: gaming industry where players trade with each other, by rules or on blackmarket, is common abuse target. See *Steam*.
- **Large sites**:

Even if the service is not a high value target per se, one might host information which is valuable blackmail or cyber reconnaissance material for a nation-state actor. If you are carefree you might even be stamped by an attack which was not especially targeting you. The worst case scenario is that the incident will strip everything of the service and it's unlikely there is an insurance fund which will cover this.

2.4 Background

2.4.1 Opening word: Welcome to Internet

In good and bad, Internet has been a dominant factor shaping the globalization and shrinking our planet. Internet allows us to connect to anybody in a blink of an eye. Internet allows us to have all the information in the world in our fingertips, no matter who wrote it, where and when. The advancements in communications have brought the unimaginable pace of development and research.

What makes Internet to be what it is, global and ever-present? When you plug your device to Internet, you make a social contract saying you accept any traffic coming over the wire, even if you don't like it. Internet itself doesn't discriminate. No evil bit exists in a network packet telling you maybe you should not open it, since there are no universal moral standards for evil. What might be a horrendous hack of a killing well-intentioned startup for one might be a small victory against the oppressors of a thief nation for the other.

Thus, in its heart, Internet is a lawless cyberanarchy. The hostile actors are foreign and often state sponsored - there is no legal remedy against someone who is not criminal by their own standards. Knowing the IP address of an adversary doesn't help if you cannot bring justice on the opposite side of the planet. There is no world police who would hear your case.

Internet has always been this way. However, today's threats are no longer just fiction in thrillers. The landscape of Internet security changed greatly during 2010s. As more business and personal life moves online, we have become dependent on Internet. We can receive more damage over Internet: there is more to lose in privacy, financial, business, industrial and political aspects. Those benefiting of exploiting weaknesses know this as well.

Something being greater than a justice system of any nation doesn't leave options for post-mortem appeals. Instead, one must recognize the environment where they are working and cope against the threats. One needs to resist them and one needs survive them on their own. Threat models and defences are well known. Building a hack-resistant system is possible and in the reach of anybody, as this guide shows.

However, hack-resistance does not happen with catch phrases like "our privacy policy states we follow modern security practices" or "our data is encrypted". These kind of opaque statements lack factual content - they don't tell *which* practices are followed and *how* your data is encrypted. Service users or peers cannot *find security* there. Alas, the history shows that every compromised service had statements like these.

To create a safer Internet we need a build a culture of security. The motivation towards this must come from the developers, operators and decision makers themselves. The first step is taking responsibility of what is at the stake and recognizing security is not something slapped on the top afterwards. Security is something crossing the business horizontally and weaved into it. The future lies in finding transparent, verifiable, ways to guarantee the security and educating peers and users about this. This guide aims to be an initiative to this direction.

2.4.2 Mission statement

Building responsible security culture yields to healthy Internet society. Every day more and more of our lives is transformed to online. People who create and nurture online services are acting deputies in this society. The motivation towards transparent trust and safety must start with these people and resonate through their work.

Security should be open and free: Nobody should need to pay to the ensure the state of the art security of their work. A willing person should be able to learn information and access tools needed to build safe Internet services for free.

Security should be transparent: There should be an effective way or approximation to say whether one can trust to the safety of a service. The intent of security should be verifiable facts, not just in claim,

2.4.3 Goals

There exist similar projects with partially overlapping scopes. This project is geared towards Internet services operators and business decision makers following the principles of openness on which Internet itself was built. The guide aims to be

- **Holistic**: The guide tells how to guarantee the security of the project as a whole. This includes organization practices, tools and devices, infrastructure and e.g. is not limited to one business function like software development practices.
- **Grassroots effort**: from bottom to up, from individuals to organizational processes. The advices in this guide are based on real life experiences. They are concrete, battle tested and addressing true concerns.
- **Open**: The content is open, under very permissive Creative Commons license. It is hosted on Github making editing open-ended for everybody.
- **Lightweight**: Everybody from a single person startup should be able to implement the security advices of the guide.

2.4.4 Prior art

OPSEC - Operations Security Guide greatly owes to earlier work in the field of information security. There exist few projects with partially overlapping scopes and the guide makes its best to cross reference them when possible.

- The Open Web Application Security Project (OWASP). OWASP is a non-profit foundation focusing to improve the security of the software. OWASP goes in-detail to good software development practices and offers invaluable advices for developers.
- PCI Security Standard Council, The Payment Card Industry Data Security Standard addresses information security concerns in credit card and payment processing industry. PCI DSS pursues the security from compliance enforcing viewpoint. The requirements are detailed, but the level of documentation, process management and reviews needed to implement them is not in the reach of a small business. PCI DSS is relevant for large enterprises where organizations themselves may not see profit in security improvements, but see it as an externalized cost. Thus, in this kind of industries, security considerations must be forced by external auditors.

- National Institute of Standards and Technology (NIST), Computer Security publications. NIST publishes comprehensive guides for security, catering everything from embedded devices to payment processing.
- Health Insurance Portability and Accountability Act (HIPAA) defines how US national standard for health care companies handling private information in electronic form. The text is in legal like, mainly geared toward legal advisors than the service operators or developers.

2.5 Assessing security

2.5.1 Introduction

Sometimes it's necessary to know that the organization and its services are secure in fact, not just in claim. Every privacy policy has a disclaimer saying they follow the industry security best practices. Rarely anybody clarifies what we are these best practices and how they are being followed. Furthermore the incidences history has shown that often the claims themselves are bogus.

This operations security guide is created in a manner that it can assess the security in an objective manner with little room for the interpretation. The assessment process is transparent and open. The guide consists of evaluation points which are easy to verify and have only outcomes "it's secure" or "it's not secure". You either get green light or red light - there is no room for grey area. When you say you have assessed the security based on Operations Security guide you can name the individual facts that were checked.

2.5.2 Chapters and evaluation points

The guide is divided to different chapters, each chapter reflecting different organizational or IT functionality.

The chapters consist of questions, called *evalution points*, on a specific matter. Ultimately you want to be able to answer yes to every evaluation point.

2.5.3 Applicable criteria

Some evaluation points might not be applicable for small businesses. For example, having a tool for managing server login keys might not be in the reach of a team of few members or less. Or it would be not needed at all, because only one person has access to the servers. Thus, each evaluation point contains an applicable criteria.

The following rough rules can be used to decide whether your organization should consider the evaluation point:

- **Everybody**: Everybody should do this, regardless of the mature and resources of their organization
- **Medium and large enterprises**: Organizations with more than two million yearly revenue or two million raised capital. These organizations generally have multiple people working on the service and can afford to hire a security consultancy, training and tools.

2.5.4 Learning security

If you are still starting a project, you are new to security to or generally wish to learn more about secure Internet services, this guide can serve you as openreference material. It contains relevant links and past security incidences which are cross-referenced with the evaluation points, building a picture how failing security has caused incidences. With historical background and more hands on examples we hope the guide is more entertaining to study than purely abstract study material.

2.5.5 Self assessment

Self assessment is a procedure where somebody within organization itself goes through the project and evaluates the security principles. This is to harden the system and make sure the project is prepared for developing secure services.

This guide can serve as a base for doing a self assessment. It contains a list of evaluation points a member of the organization can go through and suggest changes to the peers how the issues should be addressed.

2.5.6 Third party assessment

A third party assessment happens when a person who is not a stakeholder in the project is invited to review the project. If the reviewer has no conflict of interest, there is little reason to embellish the results, leading to more objective and accurate assessment.

The results can be published or left as an internal memo. Publishing the results is a trust building tool: if the Internet has a reason to trust the person who did the assessment, the audience learns the organization knows what they are doing and can trust their services.

A third party security assessment example

The following assessment process

- The organization signs NDA with the reviewer
- The reviewer goes through the evaluation points with the representatives of organization
- Each evaluation point is confirmed with manual inspection. If the organization claims the devices are encrypted, the reviewer checks that this is indeed in the case, probably starting from the laptop and mobile phone of the CEO.
- The reviewer gives feedback to address the issues. For example, if a two-factor authentication is not enabled on the hosting provider account, the reviewer and the representative enable it during the assessment.
- If there is sensitive material, like site source code, it is handled only offline, through USB stick. The material is copied only to a specific computer which is reset before and after the assessment. This shields reviewer itself against any potential compromises.
- The report and evaluation point notes are also written offline on the same specific computer. The evaluation point notes are never put online.
- The reviewer gives feedback to the organization including the full evaluation point notes and scores.

Publishing the assessment results

- The reviewer writes a public statement regarding the service security. The statement includes evaluation point score summaries for each chapter and written letter of recommendation.
- The organization can decide whether the statement is published. The statements are published on operationssecurity.org and the service can use operationssecurity.org badge to link to the results on their website.
- If some of the issues noted during the assessment are too cumbersome to be addresses on the spot, the organization can have the second assessment round. The reviewer comes back and goes through the remaining issues to see they were addresses and the evaluation point clears green.

2.6 Team security

This chapter discusses how to guarantee the safety and integrity of team members, credentials, devices, tools and software.

Instead of trying to exploit the service directly, the adversaries may go after team members, managers and partners working on the project. The project should aim to protect team communications, devices and authorization keys so that they are unlikely to get compromised. This involves following basic IT security practices, cyberhygiene, key management and limiting the impact of potentially leaked data.

Physical security, like door access keys and security cameras, is de-emphasized because these security aspects rarely reflect the reality of a mobile contemporary worker. Regardless of the broken physical security, the service should stay intact and uncompromised.

2.6.1 Basic security practices

Team members follow the basic IT security practices? Yes / No

Team members are trained on and aware of common cyber threats like phishing attacks and social engineering. They can identify basic attacks, like spearhead phishing emails, and do not fall victim to them.

Team members maintain cyberhygiene and do not use work devices to visit sites which might compromise the web browser. Software, both desktop and server, is maintained in up-to-date versions in which known vulnerabilities do not exist.

Applies for: Everyone

Related incidences:

- *Bitpay*

2.6.2 Dangerous file attachments

Potentially dangerous file attachments are handled securely? Yes / No

File attachments in email and chat are one of the most common attack vectors.

Rigged files may include:

- Office files (Microsoft Word, Microsoft Excel, and related)
- Flash animations
- PDF files

Dangerous communication channels include anything on which team members can be freely contacted, including:

- Email
- Skype
- WhatsApp

The desktop applications and web browser plugins opening this kind of content should be disabled. If disabling is not an option, the attachments in an email or outside team internal communication tool should be opened securely and never using the productivity applications themselves. Secure open methods include opening the file in a web-based viewer, web email preview or otherwise sandboxed tool. Furthermore, a safe version of a desktop productivity suite, which is preferably an up-to-date open source tool, should be used.

Applies for: Everyone

Related incidences:

- *Bitstamp*

Links:

- Office Editing for Docs, Sheets & Slides (Google Chrome add-on)
- Google Docs Viewer (Firefox add-on)
- LibreOffice

2.6.3 Password manager

Team members use a password manager? Yes / No

All team members use a password manager for their passwords. Randomization using a password manager is the only cognitively safe way to manage a lot of sensitive and strong passwords. Without randomized passwords, one compromised password may lead to the loss of other passwords due to password reuse or password pattern reuse.

Whether one can trust a third-party cloud-based service to store passwords is subject to discussion depending on the sensitivity of the project.

Applies for: Everyone

Related incidences:

- *LastPass*
- *Hacking Team*

Links:

- List of password managers (Wikipedia)
- KeePassX

2.6.4 Third party devices

Team members do not use third-party devices for logging in? Yes / No

If a device comes from a non-trusted party, it may contain keyloggers and other malware to record the user's actions. Such devices include Internet kiosks, school and library computers, and other free terminals.

Team members use only assigned devices for their work. Furthermore, the devices should be sourced from a reputable vendor.

Applies for: Everyone

Related incidences:

- *Asian Android phones*

2.6.5 Encrypted computers

Work computers have disk encryption? Yes / No

The permanent storage, SSD or hard disk on team members' computers is fully encrypted.

All desktop operating systems have disk encryption technology available: FileVault (OSX), dm-crypt (Linux) or BitLocker (Windows). The usage of disk encryption implies password authentication upon computer power-on and wake-up so that powered-on devices cannot be accessed.

A lost device, when encrypted, cannot lead to any kind of compromise.

Applies for: Everyone

Related incidences:

- *NASA*

Links:

- Use FileVault to encrypt the startup disk on your Mac (Apple)
- FullDiskEncryptionHowto (Ubuntu)
- BitLocker Drive Encryption Overview (Microsoft)

2.6.6 Encrypted mobile devices

Team members have disk encryption on their phones and tablets? Yes / No

A lost device, when encrypted, cannot lead to any kind of compromise. Even if the device were not to contain sensitive data per se, it could contain active email inboxes and team chats, leading to further account compromise and phishing.

The device should be protected by password and a not-easily guessable pattern or easily foolable fingerprint scanner.

Note: Having any kind of online recovery option for a forgotten device password is unsafe. In the case of a forgotten password, the device should be wiped and factory reset.

Note: Remote wiping tools give almost zero protection in the case of a lost device. It's trivial to take a mobile device offline and extract data from a powered-down device.

Applies for: Everyone

Links:

- Encrypt your data on Android (Google)
- iOS: Understanding data protection (Apple)
- How To Bypass Android Lock Screen (Übergizmo)
- iPhone 6 Touch ID Fingerprint Scanner Hacked Days After Launch

2.6.7 Minimized email usage

Email is not used for internal communications?

Email as media is insecure. Email travels plain-text through the Internet. Even if the message content itself is encrypted, the metadata is still readable.

Instead of email, closed tools and services should be used for team internal communications.

For highly sensitive projects, the communication should be contained in an in-house server.

Related incidences:

- *Bitpay*

Links:

- Email Privacy (Wikipedia)

- Modern Team Communication Tools for Developers (Stefan Mayer)

2.6.8 Two-factor authentication on email

Team members' work and personal email accounts require two-factor authentication to log in.? Yes / No

Inboxes contain sensitive information. Often email acts as the key to third-party services and infrastructure, as email is used for logging in with a forgotten password option. Thus, getting into the inbox further compromises other services.

Email is an attractive target to hack, as email is either public or easily guessable. Even if the email account is protected by a strong password, flaws may exist in the password reset process, e.g., by intercepting the voice mail of the target user. Two-factor authentication provides additional protection against such attacks.

Applies for: Everyone

Related incidences:

- *Bitpay*
- *CloudFlare*

Links:

- Two-factor Authentication List
- Google 2-Step Verification

2.6.9 Two-factor authentication for admins

Website administrators use two-factor authentication? Yes / No

Team members, support personnel and other people with administrative access to the website use two-factor authentication.

Internet services often provide an administrative site or access in which the site managers perform in-house updates, edits and other support tasks. This kind of administrative access should be available only through two-factor authentication.

If the attacker compromises a password of a team member, the attacker should not be able to get into the administrative site. Furthermore, administrative access can be limited to VPN or other well-known (office) IPs.

See also *Two-factor authentication*.

Applies for: Everyone

Links:

- Two-factor Authentication List

2.6.10 Passphrase on server login keys

The terminal access to the server requires passphrase protected key? Yes / No

Logging in to the server containing private data is allowed only with passphrase-protected key files.

The usual logging method is by SSH secure shell connection, but if alternative methods to access the server exist, the key files should be used there, too.

Using key files instead of passwords protects against brute force attacks, simple keylogging attacks, weak password attacks and such. Furthermore, the keys must be passphrase protected so that in case a key file itself leaks, it is useless to the attacker.

Note: If the hosting provider has a console, terminal or root password reset option on the server, special attention should be paid to this. It is better to either disable this feature or to make sure it is behind two-factor authentication and cannot be performed by hosting provider personnel.

Applies for: Everyone

Related incidences:

- *Linode*
- *MaxCDN*

Links:

- SSH key and passwordless login basics for developers (Mikko Ohtamaa)
- Linode Hacks (Bitcoin Thefts)

2.6.11 Two-factor authentication on server login

Terminal access to the server requires two-factor authentication? Yes / No

Logging in to the server containing private data requires two-factor authentication.

Server login is further restricted with two-factor authentication, so that even in case the computer of a server administrator is hijacked by malware, this computer cannot log in to the server without user interaction and a two-factor token from a separate device. This makes it nearly impossible to hijack the secure connection to the server unnoticed.

See also *Two-factor authentication*.

Applies for: Everyone

Related incidences:

- *Bitstamp*
- *Linode*

Links:

- SSH login with Google Authenticator TTOP two-factor
- Two-Factor-Authentication with SSH (Carsten Heesch)

2.6.12 Audited server login keys

A real-time method of maintaining and revoking keys across all servers? Yes / No

In any point of time, the administrators of the project can revoke any key used by the team. Full audit logs of key provision andAt any point in time, administrators of the project can revoke any key used by the team. Full audit logs of key provision and usage are available and stored separately.

This allows for the quick address of issues when a compromise is suspected.

Applies for: Medium and large enterprises

Related incidences:

- *MaxCDN*
- *Ashley Madison*

Links:

- Universal SSH Key Manager (SSH Communications Security)

2.6.13 Software installation from safe sources

Software is installed from known good sources? Yes / No

Pirated software is riddled with malware. Team members install software coming from legit sources only, reducing the risk that the software comes with malware.

Safe software channels include:

- App stores by operating system vendors
- Official, signed, UNIX distribution repositories
- Programming community package repositories

Basic security understanding and cyberhygiene should still be applied when installing from safe channels (e.g., Google Play is known to host several rigged applications).

Even if malware is not targeting the project itself, malware authors inspect infected computers for high-value targets and may open an attack if they notice such a successful infection.

Applies for: Everyone

Related incidences:

- *XCode*
- *SquirrelMail*

Links:

- Malware that Just Won't Give Up on Google Play (Avast)
- PEP 0458 – Surviving a Compromise of PyPI

2.6.14 Limited sensitive data access

Sensitive data access by administrators is limited? Yes / No / Not applicable

Administrative access often implies the ability to view users' private data.

When team members access private data, the access is limited in a way such that sensitive information is not exposed unless necessary for performing work. For example, social security numbers are not viewable among normal data unless the administrator chooses to explicitly show them.

See also: ref:*Authorization and permission framework*.

Applies for: Everyone

Related incidences:

- *Ashley Madison*
- *Hacking Team*
- *Patreon*

2.6.15 Logged sensitive data access

Sensitive data access by administrators is logged? Yes / No

All actions related to administrators' accessing and manipulating sensitive data are logged. This includes direct database connections, API services and other internal access methods.

In case of privacy breach claims, these logs can be used to reconstruct the scenario regarding who has been accessing or manipulating the data. Sensitive data access logs protect against insider threats. Knowing that one cannot get away without being caught discourages malicious attempts.

Access logs should be detailed as possible, including timestamp and details of performed actions. Access logging can be implemented, for example, by storing the full HTTP request logs, including POST parameters and body, from all logged-in administrators.

See also *Log server* and *User audit logs*.

Applies for: Everyone

Related incidences:

- *Ashley Madison*

2.6.16 Data scrubbing

Data dumps are cleaned of sensitive information? Yes / No

Instead of working with full production datasets, there exists a repeatable process of making a cleaned dataset with sensitive information removed from the data.

The data scrubbing process can reset:

- User email addresses
- Phone numbers, physical addresses and social security numbers
- Password hashes
- Two-factor tokens

The cleaned dataset is then given to the team members who need to analyse, test and develop against the data.

The cleaning process limits the impact of potential data leak in cases in which the data dump accidentally ends up at the third party. Furthermore, the cleaned data ensures that messages from the testing environment cannot reach actual users.

Applies for: Everyone

Related incidences:

- *Ashley Madison*
- *Patreon*

Links:

- How to Anonymize Data in a PostgreSQL Database (Michael Krenz)

2.7 Web development practices

This chapter discusses the need for web development best practices when creating sensitive Internet services.

When developing a sensitive Internet service, special attention should be paid to the security. It is very possible to build unhackable services. Accomplishing this requires discipline and security awareness from the development team.

Most application-level vulnerabilities are related to the input handling. Any Internet facing service accepts incoming traffic and user input, both good and bad. It's a social contract: when you plug in your service to the Internet, you acknowledge that anyone in the world is allowed to use the service.

HTTP and HTML were built during an era with fewer threat models and less of a need for security. Many security features are retrofitted or they are hacked on top of the original HTML. Anyone who is not an expert in HTTP, HTML and JavaScript has a lot to learn about potential attack modes. Thus, it is an imperative that the development team uses a proper *software development framework* to build the service. Usually framework authors are well-versed in security matters and have thought out and documented the proper processes for things like form submission handling, file uploads and exposing the database to the world.

2.7.1 HTTPS / TLS only

Service is HTTPS-only with security HTTP headers? Yes / No

The service is available only over an encrypted connection. A plain HTTP connection is allowed only for the initial redirect. Furthermore, the HTTP responses should include security headers like *HTTP Strict Transport Security*, *X-Frame-Option* and HTTPS-only cookies with no JavaScript access.

Encryption protects again man-in-the-middle attacks which include:

- Malware tapping traffic locally
- Compromised Wi-Fi routers
- Malicious Tor exit nodes
- Nationstate actors and mass surveillance

The *X-Frame-Option* HTTP response header prevents clickjacking attacks, though it is not related to transport security directly.

If the site loads resources from external content delivery networks (CDNs), these downloads should be marked with subresource integrity tags to prevent attacks through a compromised CDN provider.

Applies for: Everyone

Related incidences:

- *Tor*
- *Soho*
- *MaxCDN*

Links:

- Let's Encrypt, a free, automated, and open certificate authority brought to you by the Internet Security Research Group (ISRG)
- Security/Server Side TLS (Mozilla Wiki)
- Clickjacking (OWASP)
- X-Frame-Option (Mozilla Developer Network)
- Subresource Integrity (Mozilla Developer Network)
- Subresource Integrity (W3C specification)

2.7.2 Database injection

Software uses a framework for database queries? Yes / No

One of the most common web application vulnerabilities is a database injection attack. Developers are allowed to write queries by hand without properly sanitizing input going into the queries.

In most cases, the database is SQL based, providing an opportunity for SQL injections. This can be easily prevented by never constructing database statements by hand and by always using a framework to construct the queries so that all values are properly escaped. Manual SQL manipulation should be prevented by the application developers so that no room is left for human error.

Applies for: Everyone

Related incidences:

- *Sebastian*

Links:

- SQL injection in Wikipedia
- SQL injection in OWASP
- PCI DSS v3.1 requirement 6.5.1
- SQL injection hall of shame
- Exploits of Mom (XKCD)

2.7.3 Cross-site scripting (XSS)

Software is written in a manner such that there is no possibility of a cross-site scripting attack? Yes / No

A cross-site scripting attack is a way to perform actions on behalf of the user when the user views or clicks a compromised payload.

The usual cross-site scripting attack involves posting comments or files where the payload is not well-escaped HTML. The attack may target site visitors or site administrators.

XSS can be avoided by using a proper software development framework which always escapes variables in template output and does not rely on developers to manually escape variables in page templates, JavaScript or HTML JSON embeds.

Special attention should be paid to file uploads: both the file content and the file name provide an attack channel. It is recommended that user-uploaded content always be served from a separate top level domain (TLD).

Applies for: Everyone

Related incidences:

- *Facebook*

Links:

- Cross site scripting (Wikipedia)
- Cross site scripting (OWASP)
- Handling untrusted JSON safely (WhiteHat Security)
- Unrestricted File Upload (OWASP)
- Secure user uploads and exploiting served user content (Mikko Ohtamaa)

- User-uploaded content (Django security)
- Sending form data (Mozilla Developer Network)

2.7.4 Cross-site request forgery (CSRF)

Software is written in a manner that it doesn't accept cross-site requests?

Cross-site request forgery is an attack in which the JavaScript payload or link hosted on a third-party site performs an attack on behalf of the user of the targeted website.

The malicious third-party site loads JavaScript which makes AJAX requests to the target site where the user is logged in.

The software should be written using a framework which prevents HTTP POST submissions without the CSRF token. Any state-changing action (login, create, modify, delete) should not be an HTTP GET request.

Related incidences:

- *Twitter*
- *Soho*

Links:

- Cross-site request forgery (Wikipedia)
- Cross-Site Request Forgery (CSRF) (OWASP)
- Sending form data (Mozilla Developer Network)

2.7.5 Password storage best practices

User passwords and two-factor seeds are hashed and salted against bruteforcing? Yes / No

Password hashing is a method to prevent cleartext password storage.

This protects user password integrity in case the database is compromised and logins and passwords are dumped somewhere. Developers should not invent password storage schemes themselves, but should use a specialized library to do the password hashing and salting for persistent storage.

Applies for: Everyone

Related incidences:

- *Sebastian*
- *Slack*
- *LastPass*
- *Hacking Team*

Links:

- PBKDF2 (Password-Based Key Derivation Function 2) in Wikipedia
- Password storage cheat sheet in OWASP
- Password strength (Wikipedia)

2.7.6 Authorization and permission framework

Private pages and data access is protected by authorization framework? Yes / No

When protecting private data, a systematic authorization framework is used instead of ad-hoc conditions. A standardized permission check method leaves less room for human error in fragile permission check conditions.

In †he authorization framework approach:

- The same process is used in all permission checks.
- Manual conditions (ifs) are unnecessary to make permission checks, as the approach is prone to human error.
- All data is preferably private unless explicitly made public.
- The checks follow a standardized authorization pattern like an access control list or activity-based checks.

Related incidences:

- *PurseIO*

Links:

- Access Control Cheat Sheet (OWASP)
- Role-based access control (Wikipedia)
- Attribute-based access control (Wikipedia)
- Permissions and Authorization (Django)
- Pundit, Minimal authorization object-oriented design for Ruby on Rails
- MustBe, Authorization Plumbing For NodeJS

2.7.7 No caching policy

Sensitive resources are not cached? Yes / No

The front-end web server and web browsers cache pages and documents by default. Sensitive pages and downloads should have explicit no caching headers present.

Thread models include:

- A caching front-end web server may lead user sessions when the HTTP response with a private cookie is accidentally cached.
- The user device is compromised and sensitive information is extracted from the browser cache.

Generally, special attention should be paid to HTTP responses like:

- Generated image, audio, video and other media downloads
- Document downloads (Office files, PDF, CSV, TXT)

Links:

- The Security Impact of HTTP Caching Headers (SANS ISC InfoSec)

2.7.8 Non-guessable IDs

Publicly exposed ids are not guessable? Yes / No

If the service uses running counters as database primary keys, these IDs should not be exposed to the public.

Knowing the ID sequence allows the attacker to gain knowledge of the item count, weakening service security.

- If HTTP endpoints or pages lack proper permission checks, guessing the ID sequence allows the attacker to scrape private data.
- Sensitive business information, like user count or trade count, is exposed to the public.

Use a random ID generation method like the Universally Unique identifier (UUID) version 4 “random,” which provides 122 truly random bits for each ID.

Applies for: Everyone

Related incidences:

- *PurseIO*

Links:

- UUID (Wikipedia)
- URL safe UUIDs in the smallest number of characters (StackOverlow)

2.7.9 Non-public administration site

The administration site is not accessible or known to the public? Yes / No

Many common software platforms come with the default administration site in a location like */admin/*.

If administrative URLs are well known, the potential attack surface expands. The attacker can guess administration HTTP endpoints with vulnerabilities and try to exploit those.

The administration interface should be in a non-guessable, non-end-user-visible URL. Besides authorization, additional access restrictions should be placed upon the administration interface with two-factor authentication, VPN and IP restrictions (see Team security).

Applies for: Everyone

Related incidences:

- *Veeder-Root*
- *Patreon*

Links:

- Failure to restrict URL Access in OWASP

2.7.10 Whitehat program

The service has a public whitehat or security bounty program? Yes / No

A whitehat program, also known as a security bounty program, is a published guide that shows how the service deals with security researchers. The purpose of a whitehat program is to encourage legit security research to cover issues on the service and credit third parties for doing this work.

The third-party security researches usually scan the service using web security audit tools like Burp Suite and try to discover XSS, CSRF, database injection and authorization flaws.

The whitehat program usually includes information about:

- How to contact the service when reporting security issues
- What response time one should expect

- Security issue types that are eligible for bounty
- The amount of the bounty and how it is paid
- Crediting the researcher for uncovering the issue

There exist third party services facilitating the creation and management of whitehat programs (Cobalt, HackerOne).

Applies for: Medium and large enterprises

Related incidences:

- *Starbucks*
- *Coinbase*

Links:

- Cobalt
- HackerOne

2.7.11 Code reviews

Source code is reviewed? Yes / No

The team uses code review, also known as code inspection, as the best practice when merging changes.

All code going to the production should be reviewed at least one person who is not the original author of the code. Two pairs of eyes see better than one to catch possible mistakes.

Applies for: Medium and large enterprises

Links:

- Code Review (Wikipedia)
- OWASP Code Review Guide

2.8 Protecting service users

This chapter discusses protecting end users and guiding them to secure their accounts properly.

Even if team members maintain high security standards internally, malicious actors can go after the end users. For example, phishing operations target a group of users who are likely users of the service. If the end users give out their login credentials in the phishing attack, the attacker may damage these users even though the integrity of the service as a whole is not compromised.

The service should take several measures to protect its users so that even if the attacker gains access to the user's email inbox or password, the harm to the user is minimized.

2.8.1 Two-factor authentication

Service users are encouraged to use two-factor authentication? Yes / No

Two-factor authentication, a.k.a. multifactor authentication, a.k.a. secure login, is a method to ask for a one-time token from the user when logging in. The primary purpose of two-factor authentication is to protect the user from password compromise.

End users may lose their passwords through multiple threats like:

- The user device is compromised by malware and the password is keylogged or extracted from the running password manager
- The password is reused across multiple sites and one of the sites gets compromised. You can buy stock email and password lists on the black market.
- The password is given out on a phishing site (see *Trademark protection*)
- The password is extracted through a man-in-the-middle attack (see *HTTPS / TLS only*)

Two-factor authentication stops the attacker, equipped with a mere password, from accessing the victim's account.

Having two-factor authentication as an option is not enough. Users should be educated about two-factor authentication. Often users are not aware of threat models and the harm they may face because of lax security. Incentives, like reduced fees, should be applied to encourage the enabling of the two-factor authentication. From a business perspective, this can be justified as a reduced support cost of dealing with hacked account cases.

Popular two-factor authentication methods include:

- Mobile apps: Time-Based One Time Password (TOTP), Google Authenticator
- Paper codes: One time pad, HOTP, popular with European banks
- SMS and other phone-based methods
- Hardware devices: YubiKey, others

External services like Authy and Clef provide two-factor-as-a-service.

Google Authenticator is a popular two-factor mobile app. Despite the fact that the name says Google, you can use it on your own site. The application can be used offline independently from Google services. Google Authenticator is based on RFC 6238. There are multiple open-source implementations for all desktop and mobile operating systems.

Note: SMS is not deemed secure in the large scale. SMS messages are intercepted by mobile malware. SMS may travel in plain text, and various parties in the operator business chain can read them. Mobile number portability opens a vector for the attacker to gain access to the victim's phone number. SMS may not be reliable in third-world countries, thus making it not a viable option for global business.

Applies for: Everyone

Related incidences:

- *Apple iCloud*
- *Slack*
- *SMS intercepting trojans*

Links:

- HMAC-based One-time Password Algorithm (Wikipedia)
- Time-based One-time Password Algorithm (Wikipedia)
- Two-factor Authentication List
- $45k stolen in phone porting scam (SC Magazine)
- What is YubiKey?
- Google Authenticator (Wikipedia)
- Google Authenticator Project (Github)
- Authy
- Clef

2.8.2 Third-factor authentication

The login process goes through an additional check in abnormal circumstances?

The login process should perform an additional check if there is a reason to believe that the login attempt might not be genuine.

The users might not have two-factor authentication enabled. Even with two-factor authentication enabled, there is a chance that the user will give out the codes on a phishing site. In these cases, the service should detect abnormal conditions and perform additional checks before letting the login proceed.

The common criteria triggering third-factor authentication include:

- The country of the user's IP address has changed.
- The device or the web browser of the user has not been seen before, identified by a stored permacookie.

In these cases, the service should prompt the login to go through additional verification steps. This could be:

- Email confirmation
- SMS confirmation

Note: Third-factor authentication does not protect against cases in which the device of the user is compromised by malware and the service cannot differentiate between legit and malicious traffic coming from the same device.

Related incidences:

- *LastPass*
- *Blockchain.info*

Links:

- Detecting suspicious account activity (Google)
- Introducing Login Approvals (Facebook)

2.8.3 Re-authentication on sensitive actions

Sensitive actions should prompt for authentication again? Yes / No

Security-sensitive actions should ask for an additional authentication attempt. Mere logging in to the service should not enable the attacker to perform sensitive actions.

The additional authentication step can be:

- Give the password again.
- Email confirmation.
- SMS confirmation.
- Give another two-factor authentication token.

Sensitive actions may include:

- Making a withdrawal from the service.
- Sending money to another user.
- Changing password, email or phone number.
- Closing the account.

Asking for an additional authentication makes it difficult to automatize malicious actions, creating another layer of protection against phishing and XSS attacks.

Sensitive operations, like those in which money is transferred out from the service, should require a minimum of two different two-factor authentication codes: one for login and one for transfer. This makes phishing site operations, which intercept two-factor authentication codes, less robust. Users are more likely to notice bad URLs the longer they need to spend time on the phishing site. The reuse of two-factor authentication codes allows the attacker to transfer out the assets if the victim logs into the phishing site even once.

Related incidences:

- *Blockchain.info*

2.8.4 Brute force login prevention

Service login attempts are throttled in multiple ways? Yes / No

Attackers may try to brute force the logins of users. The service should have adequate measures to prevent multiple login attempts and to effectively stop them.

There are a few different brute force attack modes:

- Spearhead a brute force attack against a single high-value victim.
- Known email and known password combination list, leaked from a third-party site or bought from the black market.
- Known email and common password list, guessing the 1000 most-common passwords.

Attackers have been shown to possess thousands of IP addresses, so blocking individual IP addresses is not effective against a well-versed attacker.

To prevent brute force attacks, counter actions should include:

- Prevent multiple login attempts per user: require CAPTCA verification on second login attempt; allow only one wrong password attempt per user.
- Prevent multiple login attempts from the same IP address or network.
- Force all users to go through CAPTCHA before login if the system global login rate is abnormally high (botnet-based attack).

Relying solely on CAPTCHA to prevent brute forcing is not recommended, as the automated CAPTCHA solving success rates are counted in the tens of percents. Thus, the malicious networks should be identified and dropped.

Beside the security ramifications, a well-armed brute force logging attacker may cause denial of service, as the system is not able to handle all the login attempts.

Note: Forcing users to choose long passwords brings limited additional value. Passwords are effectively dead. It doesn't matter how complex the password is, as usually the whole password is lost due to phishing or keylogging malware. Instead, two-factor authentication should be encouraged as the primary option for increasing account security.

Applies for: Everyone

Related incidences:

- *Apple iCloud*

Links:

- Blocking Brute Force Attacks (OWASP)
- Rolling time window counters with Redis and mitigating botnet (Mikko Ohtamaa)

- reCAPTCHA
- Password strength (Wikipedia)

2.8.5 Effective session kill

When the user account is deactivated or changed, the related sessions are dropped?

If the attacker gains access to a user account, system administrators must be able to kick out the attacker. In certain security-related actions, it is also good practice to drop the sessions of the user.

Account deactivation, besides marking the user account deactivated in the database records, should also drop the active sessions which are usually stored in a separate backend like Memcached or Redis. When a user account is deactivated, all communication channels to this user must be dropped: HTTP sessions, WebSocket sessions, mobile application sessions and so on.

Furthermore, all user sessions should be dropped when the users themselves perform changes which may affect account security. These include:

- Password change
- Email address change
- Phone number change

After the change has been performed, the user must relogin to the service. This allows the users themselves to act quickly in situations in which they notice that somebody has hacked into the account, e.g., via an incoming email notification. In this case, the user is still probably logged into the system with stolen credentials and the user may hurry to change the password to kick the attacker out.

Related incidences:

- *Slack*

Links:

- Simultaneous Session Logons (OWASP)

2.8.6 User audit logs

Service retains audit logs of sensitive user actions?

All sensitive actions should be logged to a user-specific action log.

The users may or may not be able to view these log entries themselves. In the case of a user reporting a hacked account, the action log can be reviewed for swift judgement. In the case of a filed police report, due to an account hack, the user audit log can be handed to the officials.

The user audit log also serves an important role in protecting the service operator itself against fraud. For example, a user can make a frivolous claim that the user's account got hacked, then threaten to sue the service and publish the incident unless there is (incorrectly) reimbursement. In fact, the user might have just transferred out assets himself/herself to a friendly third party. The user audit logs prove the correct password and authentication codes were used to initiate the transfer and shift the responsibility to the user himself or herself.

The user audit log should include at least:

- User logins and login attempts
- Password change and reset operations
- Enabling and disabling two-factor authentication

- Email change operations
- All financial operations
- Timestamp with timezone
- IP address
- User agent

Related incidences:

- *Steam*

Links:

- Logging Sessions Life Cycle: Monitoring Creation: Usage, and Destruction of Session IDs (OWASP)
- Investigation report of the claimed security breach at LocalBitcoins

2.8.7 Account verification process

The creation of bogus accounts is prevented? Yes / No / Not applicable

In services in which it is possible to spam or harass other users, fake accounts are a common problem.

To keep the service clean, one should prevent the creation of fake and robot accounts. The cost of automated account creation should be high enough that there is no financial gain to create and use accounts for spamming. On the other hand, the account creation process should still be smooth enough that it doesn't discourage users from signing up.

Account verification is also important for anti-money laundering (AML) and know-your-customer (KYC) cases in which it is imperative to know that one is dealing with the rightful holder of the financial assets.

Common account verification methods include:

- CAPTCHA
- Email verification
- Phone verification
- Browser verification by security proxy (CloudFlare, etc.)
- IP reputation system (block countries where you have no business, block Tor and VPN IPs)
- Piggybacking the authentication mechanism of a large service (Facebook, Twitter, Google OAuth)
- Government ID verification services (available as-a-service like Jumio and Trulioo)

Please note that all of these can be defeated if the financial incentive of the attacker is high enough.

Related incidences:

- *Instagram*

Links:

- reCAPTCHA
- Dialing Back Abuse on Phone Verified Accounts
- Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse
- Priceless: The Role of Payments in Abuse-advertised Goods
- Facebook Asks Every User For A Verified Phone Number To Prevent Security Disaster (TechCrunch)
- Facebook Requesting Government ID to Unlock Accounts (TheBlaze)

- Jumio
- Trulioo

2.8.8 Flood action throttle

Actions sending messages to other users are throttled? Yes / No

When the service provides ways to message or contact other users, these actions should be throttled so that one cannot flood messaging by sending a large number of useless messages.

Example actions that should be throttled include:

- Sending messages to other users
- Sending invitation emails
- Sending SMS messages

If a malicious actor is free to send an infinite number of messages, this can be exploited for harassment. Even if the exploitation doesn't lead to direct financial gain for the attacker, the service may take a reputation hit and the brand suffers due to poor user experience.

Throttling can be done by having time window thresholds for how many messages one user can send or how many messages can be sent on a global level. If the frequency of actions exceeds the limit of what a normal person would do, the action should be disabled or the user banned.

Related incidences:

- *Coinbase*

Links:

- ‘ <Rolling time window counters with Redis and mitigating botnet (Mikko Ohtamaa)>‘_

2.8.9 Trademark protection

Is the name of the service trademarked??

A trademark helps protect against advertisement phishing.

If you type the service name into Google or a web browser address search bar, Google displays advertisements on the top of the actual search results. These advertisement can be bought out to create misleading phishing links, like *www.blockchain.com.de/wallet/login*. Normal end users cannot distinguish between phishing advertisements and actual search results.

If you have properly trademarked your service name, you can ask Google AdWords to not allow it to be used in advertisements, making advertisement phishing harder.

Google AdWords is known to take down phishing advertisements quite slowly when reported.

Applies for:

Related incidences:

- *Blockchain.info*

Links:

- AdWord Trademark Policy
- Report a phishing page (SafeBrowsing)

2.9 Infrastructure security

This chapter discusses how to protect servers and other infrastructure needed to run the service.

Infrastructure here covers everything that runs your Internet service. It is territory for system administrators, who are aware of integration and deployment details. Everything cannot be done in-house; one most likely needs to rely on third-party services for some things, and they should be accounted for, too.

2.9.1 Two-factor authentication on infrastructure services

Infrastructure services require two-factor authentication? Yes / No

If infrastructure services provide a two-factor authentication, this option is used.

Internet services often rely on third-party services for infrastructure. The infrastructure services could include:

- Server hosting
- Domain name services
- Certificates
- Transactional email
- SMS
- Proxy and CDN services, etc.

Two-factor authentication provides an additional layer of security against cases in which passwords of team members get compromised. It also gives protection against social engineering and password reset attacks which the attacker may try against the infrastructure service accounts.

Related incidences:

- *Bitly*

Links:

- Two-factor Authentication List
- Multi-Factor Authentication (Amazon Web Services)

2.9.2 Encrypted server data

Data is stored on encrypted partitions?

All data should be stored on encrypted partitions or files. In the case of unauthorized physical access or unauthorized reboot, the data cannot be compromised. Encryption should also apply for backups and other offsite files.

Data on encrypted partitions protect against:

- Hosting provider attacks or social engineering attacks against the hosting provider in which the root password is reset through single-user mode reboot
- Law enforcement attacks in which the servers are physically confiscated

Disk encryption protects data when the server is offline. All sensitive databases should reside on the partitions which are not accessible if the physical machine is compromised.

If the server is rebooted without authorization, the server should ask for a passphrase to decrypt the data partitions. The easiest way to achieve this is to have separate partitions for boot volume and data volume. By having separate "high" and "low" states, the server cannot enter into the state with access to data unless an authorized person enters a passphrase through a terminal.

All unaccounted reboots should be suspicious, as when the server is offline the boot mechanism can be compromised to record the data decryption keys.

Note: Virtual machines, like those provided by Amazon EC2 or Digital Ocean, are ultimately unsafe. It's possible to make a silent copy of a virtual machine and its disk, even if encrypted, without the authorization of the server owner. If the adversaries include law enforcement agencies and nation state actors, it is recommended to use physical servers with chassis removal detection in a locked rack.

Related incidences:

- *Bitly*
- *Linode*

Links:

- How To Use DM-Crypt to Create an Encrypted Volume on an Ubuntu (Digital Ocean)
- Duplicity, encrypting backup utility

2.9.3 Security proxy

Servers are behind security proxy?

Security proxy services act as a front end for web traffic. They mitigate attacks and accelerate site traffic.

Using a security proxy service hides the IP of your servers from the attacker, thus making denial-of-service attacks more difficult to perform. Security proxies are provided by specialized companies that maintain geographically distributed servers and a lot of bandwidth to withstand attacks. The security proxies may utilize an IP reputation system and force botnet IP addresses to go through additional CAPTCHAs before being allowed to connect.

Security proxies also mitigate legal threats against your hosting provider. Without public knowledge of who is hosting the servers, it is more difficult to take legal action against the service.

Links:

- CloudFlare
- Incapsula

2.9.4 Internal services not exposed

Internal servers, services and domains cannot be discovered through public records?

Internal services, testing and staging servers should not be exposed to the public. DNS records should not contain services which are not for public consumption.

Usually testing and staging servers have more people with privileged access. This increases the potential attack surface from compromised devices and accounts. If there is no specific reason why the server needs to be publicly accessible, it should be hidden behind VPN and not be visible in public DNS records. For internal services, run a custom DNS server or use a non-guessable secondary domain name.

A firewall should be only a secondary layer protecting human errors. The services should be configured in a manner such that they do not bind to publicly exposed IPs, but bind only to localhost or internal IPs. Furthermore, firewalling outgoing connections might be problematic, as many services today rely on third-party API service.

Within the server, any installed software is run under a non-admin (non-root) account. This limits the impact of an attack if arbitrary code execution vulnerability in the native applications is exploited. The attacker cannot leave further backdoors in the system and is able to gain only limited intel.

Related incidences:

- *Patreon*

Links:

- Nmap
- Privledge separation (Wikipedia)
- Basic access authentication (Wikipedia)

2.9.5 Traffic throttle

Throttle or ban IP addresses with excessive requests? Yes / No

Prevent denial-of-service, brute force and scraping attacks against your service by blocking clients doing excessive traffic.

Normal users and clients should be able to do only four to twenty burst HTTP requests to the service. If there is more incoming traffic and the client is not whitelisted, the client likely does not have good intentions.

A log monitoring software like fail2ban can do this with almost zero configuration for stock applications like SSH and common web servers.

Please note that IP blocking alone is not effective against adversaries with botnets and thousands of global IPs in their possession.

Note: Don't accidentally ban good known bots like GoogleBot and Bing.

Applies for: Everyone

Links:

- NGINX - throttle requests to prevent abuse (ServerFault)
- fail2ban
- Banning IPv6 addresses (ServerFault)

2.9.6 Log server

Critical logs are mirrored to a log service?

Critical log files should be mirrored to a destination where the logs can be only appended. The logs cannot be read back or manipulated.

The log service should have different access credentials from the administrators of normal systems. In case the attacker gains access to the infrastructure, this prevents the wiping or manipulating of logs. This allows robust recovery and post-mortem from potential attacks.

Applies for: Medium and large enterprises

Links:

- Amazon CloudWatch
- Creating a Centralized Syslog Server (Linux Journal)

2.9.7 Secure server-to-server connections

Server-to-server connections are secure?

Nation state actors and other capable adversaries are proven to be able to tap Internet backbone connections and data centers.

The server-to-server connections should be encrypted in a manner such that anyone tapping physical cables cannot extract any information, like raw database traffic.

The connection encryption methods are VPN and SSH tunnels.

Links:

- Room 641A (Wikipedia)
- Googlers say "F*** you" to NSA, company encrypts internal network (Ars Technica)
- Reports that NSA taps into Google and Yahoo data hubs infuriate tech giants (The Guardian)

2.9.8 Intrusion detection

Intrusion detection alerts on unexpected server activity?

Intrusion detection software monitors servers and alerts in case there is unexpected activity.

Intrusion detection is a monitoring measure that detects server compromises. Intrusion detection software monitors processes, file systems, configuration files, passwords and user databases. In case there are changes not matching the predefined ruleset, an alert is fired.

Intrusion detection cannot detect in-process compromises and tailored attack payloads. Thus, its efficiency against well-versed adversaries is questionable.

Applies for: Medium and large enterprises

Links:

- Tripwire
- OSSEC

2.10 Digital currencies and securities

This chapter discusses the security aspects of storing and handling digital currencies and securities like Bitcoin.

Digital currency services are especially attractive cybercrime targets. Digital currency transactions are anonymous, non-reversible and non-traceable. This makes stealing, laundering and liquidating digital currencies very easy for criminals. The non-reversable transaction mechanism complicates attacks, as often the services can neither chargeback lost assets nor reimburse customer losses.

Traditional credit card, debit card and wire transfer-backed transaction mechanisms are more merciful. Such transactions can be reversed, making it harder to liquidate stolen assets. Anti-money-laundering regulation ensures that it is not possible to move assets without leaving a trace for investigation. Furthermore, the institutions issuing cards and bank accounts have mechanisms to address fraud, co-operate with police and insure funds. For example, the compromise of an e-commerce site poses relatively little risk to its owners and customers unless the site was maintaining a balance in digital currencies.

Thus, services dealing with digital currencies and securities should approach security matters with tremendous seriousness. History shows that companies possessing millions of dollars of funding to address security still fail in basic execution (*Bitstamp*, *Bitpay*).

2.10.1 Cold wallet

Cold wallet maintains most assets offline? Yes / No

Cold wallet is a system in which private keys for digital assets are stored on a computer, not connected to the Internet. To move the assets, somebody has to go to the computer physically and make the transaction.

The opposite of cold wallet is hot wallet, which is continuously running on the server and accessible online. Hot wallet assets are instantly liquid; most customer facing transactions happen in the hot wallet. When the hot wallet gets "too full," assets are moved to the cold wallet to make them safe. On the other hand, when the hot wallet is drying up due to customer withdraws, it must be topped up from the cold wallet.

In case of a service compromise, the attacker can only drain the hot wallet, severely limiting the amount of potentially stolen funds.

Applies for: Everyone

Related incidences:

- *Linode*
- *Bitstamp*
- *Cryptoine*

Links:

- Cold storage in Bitcoin Wiki

2.10.2 Race condition prevention

A systematic development method prevents race conditions? Yes / No

A systematic development method is applied to all financial transactions so that race conditions cannot compromise transaction integrity. Otherwise exploiting the race condition allows the attacker to manipulate account balances.

For all financial transactions:

- Optimistic database-level transaction isolation is applied or...
- Pessimistic application level locks are applied.

The software should be developed in such a manner that there is only one function to make transfers out from the system or within the system. This function has a locking mechanism such that simultaneous transactions from the same account cannot compromise the atomicity, leading to double top up, double withdraw or account overdrawn.

Related incidences:

- *Cryptoine*
- *Starbucks*

Links:

- Lock (database) (Wikipedia)
- Race condition (Wikipedia)
- Atomicity (Wikipedia)
- Transaction Isolation (PostgreSQL)
- How I stole roughly 100 BTC from an exchange and how I could have stolen more

2.10.3 Transaction verification

Withdraws are verified by heuristics?

Withdraws are verified by heuristics so that unusual outgoing transactions need another round of authorization from the customer or human interaction from the support team.

Outgoing transaction verification provides an additional layer of protection against asset theft:

- Customer withdraws are verified. If the parameters of the transaction do not match prior customer activity and a malicious withdraw is suspected, the customer must reauthorize the transaction (see *Third-factor authentication*).
- How wallet drain attacks are prevented, as the heuristics would detect such and stop them.

Transaction verification is usually implemented as a multi-signature service with a third party. A third party holds one key required to make the transaction. When a transaction is created, the third-party service checks the transaction parameters against known good rules. If the transaction looks okay, the third-party service signs its part of the transaction. Because the third party is independent and specialized in the transaction verification process, it is unlikely that the attacker would manage to compromise it, too.

Related incidences:

- *Bitstamp*
- *Linode*

Links:

- BitGo
- BitGoD (Github)

2.10.4 Multisignature for major withdraws

A minimum of two parties are required for a large withdraw? Yes / No

A sole person alone should not be able to compromise the cold wallet or customer assets. Requiring authorization from two different people makes it less likely that one person disappears with all the customer assets.

Digital currencies provide a multi-signature mechanisms. A withdraw action can be set to require minimum of two different parties to confirm it. Such a mechanism should be used any time a large fraction of assets are moved e.g. topping up the hot wallet from the cold wallet.

Applies for: Medium and large enterprises

Related incidences:

- *Bitpay*

Links:

- Multisignature (Bitcoin Wiki)

2.10.5 Proof of solvency

The service is able to perform Proof-of-solvency? Yes / No

Proof of solvency (PoS) is a scheme designed to let users verify the solvency of online websites which accept Bitcoin deposits in a way that doesn't compromise the privacy of users.

Proof of solvency is used as a public proof to verify that the service does not run as a fractional reserve, e.g., some of the customer assets could not be withdrawn on a given moment. It is mostly used by Bitcoin exchanges to prove that they still have the assets that the customers have deposited.

The current proof of solvency schemes usually involves:

- A (merkle tree) hashing scheme
- A third-party auditor
- A public statement

The third party verifies that the exchange was controlled in all given Bitcoin addresses and that they have more unspent Bitcoins than claimed total customer assets.

The service should be able to perform a proof of solvency audit, at least internally.

Applies for: Medium and large enterprises

Related incidences:

- *Mt. Gox*

Links:

- Proving Your Bitcoin Reserves (Zak Wilcox)
- Proof of Solvency specification
- ‘ <Bitfinex Passes Stefan Thomas’s Proof Of Solvency Audit ()>‘_

2.11 Security incidence reference

This chapter contains references to historical security incidences, why they happened, the implications and what operational security measurements could have been taken to prevent them.

Attention: *All Internet service incidences listed here could have been avoided by following the security practices presented in this guide.*

Some of the incidences are not directly related to a particular Internet service, e.g. SMS intercepting, but the case reflects the associated security risk it may pose to any Internet service and its user.

2.11.1 Incidences

Number of incidence summaries: **31**

Compromised user accounts: **39.29M**

Lost assets: **457.19M USD**

Bankcrupted companies: **1**

Fired employees: **0**

2.11.2 Indicent index

Apple iCloud

Date: 2014-09-01

Apple iCloud service was subject to login brute force attack leading to the leak of celebrity private photos.

Apple did not follow the security best practices to prevent brute forced login attempts. Find my iPhone, a part of iCloud services, allowed unlimited login attempts. This allowed the attackers to guess simple passwords for known email addresses.

Later the private photos of victims, most of them being celebrities, were leaked in Internet.

Apple did not apologize.

Related evaluation points:

- *Two-factor authentication*
- *Brute force login prevention*

Links:

- Apple Media Advisory - Update to Celebrity Photo Investigation
- Apple patches 'Find My iPhone' exploit (ZDNet)
- Find My iPhone exploit may be to blame for celebrity photo hacks (Engadget)
- Was iCloud vulnerable... (Quora)

Ashley Madison

Date: 2015-07-01

Compromised user accoutns: **36M**

Ashley Madison, billed as an extramarital affairs service, got comprehensibly compromised.

A Canadian company Avid Life Media was running a dating site for married people. All the company data was leaked to public, including the production database, internal emails, discussions and marketing memos.

"In July 2015, a group calling itself "The Impact Team" stole the user data of Ashley Madison, a commercial website billed as enabling extramarital affairs. The group copied personal information about the site's user base, and threatened to release users' names and personally identifying information if Ashley Madison was not immediately shut down. On 18 and 20 August, the group leaked more than 25 gigabytes of company data, including user details." (Wikipedia)

"Because of the site's policy of not deleting users' personal information – including real names, home addresses, search history and credit card transaction records – many users feared being publicly shamed." (Wikipedia)

As the writing of this it is not yet public information how the compromise happened. A black hat hacker group called "The Impact Team" distributed the data dumps. What is missing is that how the group get their hands on the data in the first place. However the extend of the data dump, including marketing documents, C-executive emails and and PayPal accounts suggest that this was either an inside job or the hackers spend a lot of time in the Avid Media internal network. The CEO of Avid Life Media says the breach was by an insider who was not an employee, but the claim has not been publicly confirmed.

The incident exposes some frivolous business practices of Avid Media, like very high generated fake and bot profile ratio.

After the incident, men who were members on the site started to receive blackmail threats for exposing the affairs to their spouses unless the blackmailer is paid in Bitcoin.

Related evaluation points:

- *Audited server login keys*
- *Limited sensitive data access*
- *Logged sensitive data access*

- *Data scrubbing*

Links:

- Ashley Madison data breach (Wikipedia)
- Who Hacked Ashley Madison? (Krebs on Security)
- Second Ashley Madison dump prompts more inside-job speculation (The Register)
- Ashley Madison CEO says hack was an inside job (Business Insider)
- An Insider Betrayed Ashley Madison (TechNewsWorld)

Asian Android phones

Date: 2015-09-01

Various Android phones manufactured in Asia ship with malware preinstalled. This includes brands available in western markets like Huawei, Lenovo and Xiaomi.

G DATA security experts discovered over 26 Android phone models which are sold having malware preinstalled. Supply chain companies, operators or manufacturers themselves are suspected of planting the malware. The attacker siphons the user data and then resells it on the black markets to substitute the phone price. The malware is usually hidden in a legitimate app which is manipulated to contain malware as an add-on.

The incident puts the Asian phone companies into very bad light, as it hints their devices should not be trusted in sensitive projects.

Related evaluation points:

- *Third party devices*

Links:

- G DATA Releases Mobile Malware Report for the Second Quarter of 2015
- Chinese Android smartphones now shipping with pre-installed malware

Bitly

Date: 2014-05-08

Compromised user accoutns: **Not disclosed**

Bitly, an URL shortening service, had their unecrypted backups compromised.

Bitly is a URL shortening service. The users can log in with their Facebook and Twitter accounts and shorten URLs. Bit.ly also offers enterprise services where the customers are allowed to use custom domains.

In the incidence, the attacker gained access to offsite unencrypted database backups. It is assumed the database contained (OAuth) tokens to take actions in Facebook and Twitter on behalf of the user.

“On May 8 [2014], the Bitly security team learned of the potential compromise of Bitly user credentials from the security team of another technology company. We immediately began operating under the assumption that we had a breach and started the search for all possible compromise vectors.” (More detail)

“They observed that we had an unusually high amount of traffic originating from our offsite database backup storage that was not initiated by Bitly.” (More detail)

“We audited the security history for our hosted source code repository that contains the credentials for access to the offsite database backup storage and discovered an unauthorized access on an employee’s account. We immediately enabled two-factor authentication for all Bitly accounts on the source code repository and began the process of securing the system against any additional vulnerabilities." (More detail)

"Hashed passwords were exposed but plain text passwords were not. All passwords are salted and hashed. If you registered, logged in or changed your password after January 8th, 2014, your password was converted to be hashed with BCrypt and HMAC using a unique salt. Before that, it was salted MD5." (More detail)

After the incidence users were asked to reset their passwords and all OAuth tokens were revoked.

The authoritative report "More detail", by Bitly, is now taken down (dead URL is http://blog.bitly.com/#85169217199).

Related evaluation points:

- *Two-factor authentication on infrastructure services*
- *Encrypted server data*

Links:

- Bitly users must change passwords, account credentials might have been compromised
- More detail (Bitly blog in the Wayback machine)

Bitpay

Date: 2015-09-17

Assets stolen: **1.8M USD**

Bitpay, a popular Bitcoin payment gateway, lost 1.8M USD in phishing attack.

Bitpay's Chief Financial Office, Bryan Krohn, received an email from a business partner email address having a link pointing to a phishing site. Krohn proceeded to give out his email account credentials on the site. The attacker logged into Khrohn's email and gained intel from the past conversations in the inbox.

Then the attacker send email to CEO Stephen Pair and executive chairman Tony Gallippi to authorize payments to "a customer wallet". There were four different payments, total of 5,000 BTC, 5,000 BTC, 3,000 BTC and 1,000 BTC. For the last payment, the CEO Pair asked confirmation from Krohn via email if the request was genuine. The attacker, still in control of the Krohn's email account, replied.

The C-level executives of Bitpay become aware of the attack when one of the real customers was CC'd in the email conversation. This customer contacted Bitpay that they had not asked for the payment.

Bitpay tried to get its insurer to cover $950,000 of the loss, but in June 2015 the insurer declined to pay. Bitpay is now suing the insurer.

Related evaluation points:

- *Basic security practices*
- *Minimized email usage*
- *Two-factor authentication on email*
- *Multisignature for major withdraws*

Links:

- Atlanta's Bitpay got hacked for $1.8 million in bitcoins (Atlanta Business Chronicle)
- BitPay Sues Insurer After Losing $1.8 Million in Phishing Attack (CoinDesk)

Bitstamp

Date: 2015-01-04

Assets stolen: **5M USD**

Bitstamp, one of the largest Bitcoin exchanges, lost 5M USD customer assets due to a breach.

Bitstamp (bitstamp.net) is one of the largest Bitcoin exchanges in the world. Their system was breached 4th January 2015. 18000 Bitcoins, worth of 5M USD by the time, were stolen. After the breach Bitstamp rebuilt their server infrastructure and partnered with BitGo, a transaction policy and clearing party. Bitstamp never commented the cause of the incident in public.

Later a confidential memo by the general counsel of Bitstamp leaked and shed some light on the events. Though the authenticity of the memo is not confirmed by Bitstamp, there is all reasons to believe it is genuine.

Below are some direct questions from this memo.

"On 4 November 2014, Mr Merlak [CTO of Bitstamp] was contacted by Skype account punk.rock.holiday from IP address (94.185.85.171). The gambit for this phishing attack was to offer Mr Merlak free tickets to Punk Rock Holiday 2015. (Merlak is keen on punk rock and has played in a band.) ... Over a period of approximately five weeks, four more Bitstamp employees received similar highly targeted phishing attacks, each tailored to individual interests." (Bitstamp Incident Report)

"On 9 December 2014, Bitstamp's Systems Administrator, L.K., received a phishing email to his Gmail account. Unlike some of the others targets, K did have access to Bitstamp's hot wallet. The email header had been spoofed to appear as if it had been sent from konidas@acm[.]org, although it was actually received from a Tor exit node. The sender was offering Mr. K the opportunity ... as part of this offer, the attacker sent a number of attachments. One of these, UPE_application_form.doc, contained obfuscated malicious VBA script. When opened, this script ran automatically and pulled down a malicious file from IP address 185.31.209.145, thereby compromising the machine." (Bitstamp Incident Report)

"On 29 December 2014, SSH logs show that Mr K's account logged in to X and the Y server at the data centre. On this occasion, Mr K was certain that these log-ins were not made by him, and must therefore have been the attacker. Analysis indicates that the attacker accessed X, where the wallet.dat file was held, and the Y server, where the passphrase for the Bitcoin wallet was stored, before data was transferred out of both servers to IP address 185.31.209.128, which is part of a range owned by a German hosting provider." (Bitstamp Incident Report)

"Two-factor authentication was not required to access the data centre from Mr K laptop while it was logged in to the office network" (Bitstamp Incident Report)

Even though Bitstamp followed high level security procedures, team members and their working devices were vulnerable. By allowing unsafe Office software suite to view an external document instead of a web based viewer, the laptop of a high value victim was compromised. Even though the server required two-factor authentication this was disabled in certain circumstances for the convenience. At the time of the incident Bitstamp did not utilize any kind of fraudulent withdraw detection mechanism. Any of these three factors would have stopped the attacker.

After the incidence Bitstamp rebuild the server infrastructure and opted in to BitGo, a third party Bitcoin transaction verification service. The Bitcoin losses were reimbursed by Bitstamp.

Related evaluation points:

- *Dangerous file attachments*
- *Two-factor authentication on server login*
- *Cold wallet*
- *Transaction verification*

Links:

- Hackers steal $5 million from major bitcoin exchange (Fortuna)

- Major Bitcoin Exchange Bitstamp Goes Offline After Possible Hack (WIRED)
- Bitstamp Incident Report (Office of Inadequate security)

Blockchain.info

Date: 2015-06-01

Assets stolen: **60k USD**

Blockchain.info, a popular Bitcoin wallet service, has been a target for Google AdWords phishing campaigns multiple times over the years 2013-2015.

The attackers buy the top spot of the Google AdWords for words "blockchain info". The advertisement points to a phishing site.

The users arrive to Google search results when typing "blockchain info" or similar input to the web browser URL bar. When viewing the result page, the end users don't understand the difference between the phishing ad and the legit search results.

The link in the phishing advertisement has been genuine-like *www.blockchain.com.de/wallet/login* or totally bogus pointing to some random static site hosting service.

When the victim logs in to the phishing site, which have been made to look like blockchain.info, the attackers steal the credentials. Then the attackers proceed to empty the blockchain.info wallet of the victim.

Blockchain.info does not disclose phishing data or Bitcoins lost to phishing. They do not reimburse the users' losses. The reports from individual users on popular forums suggest at least 25 BTC have been stolen this way.

Related evaluation points:

- *Third-factor authentication*
- *Re-authentication on sensitive actions*
- *Trademark protection*

Links:

- Blockchain.info Phishing site 1st result on Google, paid ad. WT*! (Reddit)
- Blockchain info phishing adverts on Google front page. This lasts already few months and Blockchain does not care at all! (Reddit)
- Blockchain.info gets tough on phishing (CoinDesk 2013)

CloudFlare

Date: 2012-06-04

CloudFlare, a CDN and security proxy company, had their service compromised due to flawed password and two-factor authentication reset process in GMail.

Matthew Prince, the CEO of CloudFlare, had his personal Google email account hacked. The account was protected by two-factor authentication.

Google offers two-factor authentication on their web based email a.k.a. GMail. Two-factor authentication should protect against cases where the attacked somehow gains access to the account password. In this case, the two-factor authentication is believed to be reset through social engineering AT&T customer support. Prince's voicemail message was modified by the attacker in order to receive and record an automated phone call from Google with a audible code that could be used to reset his account credentials.

The personal email account of Prince was the recovery email for Google Apps for Business. After gaining the access to Apps, the attacker could read some transaction email traffic, including password reset emails, which was BCC'ed to CloudFlare team. BCC email feature was for error diagnostics. The attacker performed password reset on 4Chan.org account, grabbed the password reset email, logged in to 4Chan account and then was able to redirect all 4Chan.org traffic to a page under the control of the attacker.

After the incidence Google changed their password and two-factor authentication reset procedures.

Related evaluation points:

- *Two-factor authentication on email*

Links:

- The Four Critical Security Flaws that Resulted in Last Friday's Hack (CloudFlare)
- Google Two-Factor Authentication Flaw Exposed Google Apps Customers (SecurityWeek)

Coinbase

Date: 2014-04-01

Coinbase has a Request money feature which sends email to any email address through Coinbase service.

Coinbase did not throttle sending actions allowing anyone to send infinite number of Request money emails. Furthermore the feature exposed if any email had an associated account on Coinbase service.

The security researcher reported the issue to Coinbase through whitehat program. Coinbase marked the issue as "WONTFIX" one month later. It was not until a publicly demonstrated exploit when Coinbase took action. In this point, it had become common prank and harrashment to send these emails.

Coinbase started to throttle the action. The company received PR damage as the user community did not find the initial response of Coinbase sufficient and questioned the security of Coinbase as a whole.

Related evaluation points:

- *Whitehat program*
- *Flood action throttle*

Links:

- Coinbase design allows for mass, targeted phishing of its users (Shubham Shah)
- Coinbase denies security breach, defends spamming-friendly features (Help Net Security)
- Update on Coinbase Data Security

Cryptoine

Date: 2015-04-04

Assets stolen: **1800 USD**

A race condition existed in the service of Cryptoine, now defunct Bitcoin exchange.

The race condition allowed the attacker to drain hot wallets of multiple cryptocurrencies of the service.

This damage caused the closure of the exchange. However the service was not popular, having only around 6 Bitcoins in their hot wallet.

Related evaluation points:

- *Cold wallet*

- *Race condition prevention*

Links:

- Cryptoine.com HACKED [race condition bug] [exchange closed] (BitcoinTalk)
- Bitcoin exchange Cryptoine hacked (ZDNet)

Facebook

Date: 2011-04-11

Facebook status update functionality did not properly escape HTML.

It was possible to post HTML content which was not properly sanitized. The malicious HTML snippet could load and execute JavaScript code in the wall comment. This allowed the attacker to create a worm which propagated through Facebook walls.

The root cause is that PHP's built-in *parse_url()* function does not properly check for malformed URLs. The issue still exists in PHP today and is only addresses in the documentation.

Related evaluation points:

- *Cross-site scripting (XSS)*

Links:

- Recent Facebook XSS Attacks Show Increasing Sophistication
- Bug #54600

Hacking Team

Date: 2015-06-05

Hacking Team was a company selling offensive intrusion and surveillance capabilities to governments. Hacking Team got compromised, having 400GB of internal data leaked.

The stolen information was likely accessed and downloaded via the compromised computers of Christian Pozzi and Mauro Romeo, two Hacking Team's sysadmins.

Though the company worked in highly sensitive business, the leaked data demonstrated lax security standards of Hacking Team operations. Weak password policies, lack of sensitive data access limitations and bad software development practices. For example, the customer software contained a hidden switch to disable it. This switch was exposed in the leak, forcing all the customers to stop using the software.

As the writing of this the attacker and the attack vectors are still unknown.

Related evaluation points:

- *Password manager*
- *Limited sensitive data access*
- *Password storage best practices*

Links:

- Hacking Team (Wikipedia)
- Hacking Team hacked, attackers claim 400GB in dumped data (CSO Online)
- Hacking Team goes to war against former employees, suspects some helped hackers (Ars Technica)
- Hacking Team's KillSwitch – Disabling the Galileo RCS remotely and silently (4Armed)

Instagram

Date: 2014-12-08

Instagram deleted millions of fake accounts.

Due to relaxed account creation process, a large proportion of Instagram accounts were fake robot accounts. The fake accounts can be used as fake followers or to send spam. Social media and PR companies often buy fake followers to inflate their campaign success rates.

It can be speculated that even if being aware of the issue Instagram was not in rush to delete the fake accounts or fix the account creation process before the acquisition by Facebook. Larger user count looks better in the valuation.

In December 2014 Instagram decided it's time delete the fake accounts. Some celebrities lost up to 90% of their followers. Instagram's own Instagram account lost 30% of its followers.

Related evaluation points:

- *Account verification process*

Links:

- Instagram mass-deletes spam accounts, users freak out
- Chaos Ensues As Instagram Deletes Millions Of Accounts

LastPass

Date: 2015-06-10

Compromised user accoutns: **Not disclosed**

A popular password management service, LastPass, has its user database compromised.

LastPass account email addresses, password reminders, server per user salts, and authentication hashes were compromised. LastPass did not disclose why the compromise happened.

The salted user master passwords where exposed to the attacker. A weak master password could lead to the compromise of the whole password vault of a user. All users were prompted to change their master passwords. LastPass does third factor authentication on its users, claiming this could have protected the potential victims.

LastPass post-mortem has a lot of talk about what they are going to do increase the security. However the actual root cause of incidence is not addressed.

The incident shows trusting a third party password management service is not always a good idea.

Related evaluation points:

- *Password manager*
- *Password storage best practices*
- *Third-factor authentication*

Links:

- LastPass Security Notice
- Hack Brief: Password Manager LastPass Got Breached Hard

Linode

Date: 2012-03-01

Assets stolen: **230k USD**

A vulnerability in the customer support system of Linode, a hosting provider, was used to obtain administrator access to the servers of multiple Bitcoin services.

Linode offers budget virtual servers for hosting. Several Bitcoin companies where hosting their site at Linode back in 2012.

The attackers exploited a vulnerability in the Linode customer support interface. The web interface for server maintenance offered a root password reset through a single user mode reboot. The attackers used this feature to the servers and root passwords. Then the attackers proceeded to logging in the servers and drained the hot wallets of victim Bitcoin services. 230k USD worth of Bitcoins were stolen.

Linode has not disclosed what kind of vulnerability it had.

Related evaluation points:

- *Passphrase on server login keys*
- *Two-factor authentication on server login*
- *Encrypted server data*
- *Cold wallet*
- *Transaction verification*

Links:

- Cloud Service Linode Hacked, Bitcoin Accounts Emptied (ThreatPost)
- Linode Hacks (Bitcoin Thefts)
- Customer support transcript with Linode (Marek Palatinus)

MaxCDN

Date: 2013-07-02

MaxCDN, a content-delivery network service had their servers compromised. MaxCDN is running bootstrapcdn.com, a CDN download for popular Bootstrap front end framework.

The vendor of MaxCDN had laid off a support engineer having access to the servers where BootstrapCDN runs. The credentials of the support engineer were not properly revoked. The attackers had gained access to these credentials. The attackers rebooted the server into single-user mode, changed the root password, and SSH'd into the server. Bootstrap JavaScript files were modified to serve an exploit toolkit.

Bootstrap is widely deployed and CDN option is one of the recommended ways to include Bootstrap on your website. BootstrapCDN gets a lot of downloads. Thus, the attack payload was served to tens of thousands of visitors in short period of time.

Related evaluation points:

- *Passphrase on server login keys*
- *Audited server login keys*
- *HTTPS / TLS only*

Links:

- BootstrapCDN Security Post-Mortem

Mt. Gox

Date: 2014-02-01

Compromised user accoutns: **190k**

Assets stolen: **450M USD**

Mt. Gox, the most popular Bitcoin exchange in 2013, declared it had lost all the assets of the customers.

"Mt. Gox was a bitcoin exchange based in Tokyo, Japan. It was launched in July 2010, and by 2013 was handling 70% of all bitcoin transactions. In February 2014, the Mt. Gox company suspended trading, closed its website and exchange service, and filed for a form of bankruptcy protection from creditors called minji saisei, or civil rehabilitation, to allow courts to seek a buyer. In April 2014, the company began liquidation proceedings.[4] It announced that around 850,000 bitcoins belonging to customers and the company were missing and likely stolen, an amount valued at more than $450 million at the time.[5][6] Although 200,000 bitcoins have since been "found", the reason(s) for the disappearance—theft, fraud, mismanagement, or a combination of these—are unclear as of March 2014." (Wikipedia)

Th ex-employees claimed that the owner of Mt. Gox, Mark Karpeles, was embezzling customer funds. It is also suspected Mt.Gox had lost customer Bitcoins and funds during the years and was running on fractional reserver for long time.

Related evaluation points:

- *Proof of solvency*

Links:

- Mt. Gox (Wikipedia)
- Mt. Gox bank fraud investigation (CCI AG)
- MtGox employee: Bitcoin boss Mark Karpeles was a 'maverick mindf**k'

NASA

Date: 2012-11-15

Compromised user accoutns: **10k**

NASA lost a laptop containing data on 10,000 users.

Personally identifiable information of at least 10,000 NASA employees and contractors remained at risk of compromise.

The laptop did not have whole disk encryption, making it possible for the thief to access all the data.

The incident prompted an immediate agency-wide initiative to implement full disk encryption on all NASA laptops.

Related evaluation points:

- *Encrypted computers*

Links:

- NASA breach update: Stolen laptop had data on 10,000 users

Patreon

Date: 2015-09-01

Compromised user accoutns: **2.3M**

Patreon, a crowdfunding site, had their development server compromised, leading to the loss of production data and source code.

Email addresses, private messages and bcrypt-encrypted passwords of 2.3 million users were lost with 15 gigabytes of data. The data was copied off from Amazon AWS development server. The development server contained full production dataset without any scrubbing.

The development server was running a debug interface connected to the Patreon Python web application (Werkzeuk on Flash). There was no authentication for the debug interface access. Anyone could connect to it and have full access to the system.

Patreon claims social security numbers and tax information were encrypted in the database, but does not clarify if the attacker gained the keys to decrypt this information.

Related evaluation points:

- *Limited sensitive data access*
- *Data scrubbing*
- *Non-public administration site*
- *Internal services not exposed*

Links:

- Important Security Notice from Patreon
- Gigabytes of user data from hack of Patreon donations site dumped online (Ars Technica)
- How Patreon got hacked – Publicly exposed Werkzeug Debugger (Detectify Labs)

PurseIO

Date: 2015-07-31

Guessable ids and sequences allowed the researcher to scrape private order data from PurseIO service.

PurseIO is a service where one can pay in Bitcoin for somebody to ask him or her to make an Amazon order on the behalf of the payer.

PurseIO AJAX call endpoint had guessable id sequence, allowing the researcher to scrape semi-private data.

Related evaluation points:

- *Authorization and permission framework*
- *Non-guessable IDs*

Links:

- Purse.io Data Spelunking

SMS intercepting trojans

Date: 2015-09-01

Multiple malware and trojan programs have been observed to intercept SMS two-factor authentication codes.

The trojans usually target banks and popular financial services. Malware is Android ecosystem issue. Though devices running other operating systems, especially jailbroken iOS, have been found infected.

When the user receives two-factor authentication codes over SMS the codes are forwarded to the attacker. The malware also does keylogging and intercepts logins and passwords of popular services.

Related evaluation points:

- *Two-factor authentication*

Links:

- New Banking Trojan Targets Android, Steals SMS (ThreatPost)
- Zeus Banking Trojan Hits Android Phones (InformationWeek)

Sebastian

Date: 2013-10-23

Compromised user accoutns: **Not disclosed**

Assets stolen: **100k USD**

A hacker group TeamBerserk claimed to have stolen more than 100k USD via SQL injection attack from Sebastian, a Californian based ISP.

The attackers downloaded the list of ISP's customers, their email addresses and passwords in clear text, through a SQL injection attack. The attackers then exploited the fact the users recycle the same password through popular services like GMail, PayPal, CitiBank, etc. The attacker used the credentials to log in to these services and empty the accounts.

The attack was demonstrated on a video uploaded to MEGA (now defunct).

Tom Dominico, the marketing and business development manager for Sebastian, told "We are aware of the claims that our system has been compromised. We have checked with our service providers and their records indicate that no such attack has occurred. We take the security of our customer's personal information very seriously and are constantly working to keep them safe from online threats."

Related evaluation points:

- *Database injection*
- *Password storage best practices*

Links:

- Hacker group claims to have looted $100k via SQL injection attack (SC Magazine)
- Hacker stole $100,000 from Users of California based ISP using SQL Injection (The Hacker News)

Slack

Date: 2015-03-01

Compromised user accoutns: **500k**

The database of Slack, a popular team communication tool, was leaked.

Slack is a popular team communication tool among software development companies. The database of Slack got compromised, leading to the exposure of salted passwords.

Slack did not disclose how the attackers got access to their database.

After the breach Slack detected suspicious activity targeting some of its customers. Slack reseted the passwords for these customers. Furthermore, after the incident, Slack enabled two-factor authentication and kill switch as options for its users. Two-factor authentication was not an option before Slack got hacked.

Whether two-factor authentication effectively stops the attackers in the case of database breach is a subject to discussion. If the salted passwords are compromised you usually also lose the two-factor authentication tokens stored in the same database.

Related evaluation points:

- *Password storage best practices*
- *Two-factor authentication*
- *Effective session kill*

Links:

- March 2015 Security Incident and the Launch of Two Factor Authentication
- Slack enables two-factor authentication following security breach

Soho

Date: 2015-05-16

Compromised user accoutns: **300k**

Hackers hijack 300,000 SOHO routers with man-in-the-middle attacks.

SOHO routers were infected via drive-by download attacks and malvertising on popular websites. The initial drive-by attack exploited a CSRF flaw in the router administration page. When a victim behind the router visited a malicious site, a JavaScript payload reconfigured the routers.

The attackers modified the routers DNS settings so that everybody from the router network could be redirected to a malicious site. This puts all sensitive transactions made from the network to risk.

Related evaluation points:

- *HTTPS / TLS only*
- *Cross-site request forgery (CSRF)*

Links:

- Malware don't need Coffee
- Exploit Kit Using CSRF to Redirect SOHO Router DNS Settings
- Hackers hijack 300,000 SOHO routers with man-in-the-middle attacks

SquirrelMail

Date: 2007-12-18

SquirrelMail, a popular self-hosted web email platform, had its source code repository poisoned.

SquirrelMail was a popular self-hosted web email application during 00s. The attacked managed to compromise the download repository and modify the packaged application. The distributed package was modified to include a remote file inclusion bug allowing the attackers to execute arbitrary code on any compromised SquirrelMail installation, leading to the compromise of the server running SquirrelMail.

The attack was detected due to mismatching package MD5 signatures.

The compromise is believed to happen through a hacked maintainers account, but it is not known how it was hacked in the first place.

Related evaluation points:

- *Software installation from safe sources*

Links:

- SquirrelMail Repository Poisoned with Critical flaw (Sûnnet Beskerming)
- SquirrelMail Repository Poisoned

Starbucks

Date: 2015-05-21

A researcher was able to manipulate gift card account balances on Starbucks by exploiting a race condition in its gift card value-transfer protocol.

By doing two gift card value transfers at the same time, the researcher was able to duplicate the transfer and duplicate the balance on the accounts of the researcher.

The researcher found it difficult to disclose the exploit to Starbucks. Starbucks customer support did not know how to contact the service developers. Later Starbucks did not thank the researcher for his efforts, but too a hostile stance against security research.

Related evaluation points:

- *Whitehat program*
- *Race condition prevention*

Links:

- Hacking Starbucks for unlimited coffee (Egor Homakov)
- Race Condition Exploit in Starbucks Gift Cards (Schneier on Security)

Steam

Date: 2015-07-25

A flaw in a password reset procedure allowed logging in to any account of Steam, a popular video game platform.

A bug in Steam, a popular gaming platform and store by Valve, allowed to reset the password of the user without entering the verification token send via email. Valve claims that accounts with two-factor authentication were protected, though there are some conflicting third party claims related to this.

The attackers, mostly people who harassed online gaming community celebrities, used forget password feature to request a Steam account password reset. But instead of reading the confirmation email, one could submit empty (“”) verification code and it passed as valid.

Valve forced the users with suspected malicious password reset to go through additional password reset procedure.

Related evaluation points:

- *User audit logs*

Links:

- Steam accounts hacked during security lapse "bug" (TrustedReviews)
- Valve patches huge password reset hole that allowed anyone to hijack Steam accounts (ComputerWorld)

Tor

Date: 2014-01-22

Security researches detected Tor exit nodes performing man-in-the-middle attack on the traffic.

Tor is a layered network to obfuscate the source of the traffic i.e. hide your tracks. It is used by criminals, privacy advocates and security researchers. Tor relies on exit nodes computers where the traffic comes out from Tor network and connects to normal Internet.

Malicious Tor exit nodes where intercepting the traffic. They performed HTTP traffic snooping, HTTP -> HTTPS redirection interception and HTTPS man-in-the-middle with self-signed certificate. There are known cases where the victim accepted the self-signed HTTPS certificate even though Firefox-based Tor browser presented a red warning screen and stating one should not proceed.

Tor developers are working to make Tor network and Tor browser to mitigate this kind of attacks.

Related evaluation points:

- *HTTPS / TLS only*

Links:

- What the "Spoiled Onions" paper means for Tor users
- Scientists detect "spoiled onions" trying to sabotage Tor privacy network

Twitter

Date: 2010-09-26

Twitter allowed to post a tweet using a HTTP GET request.

There was no CSRF check for posting a tweet this way. The attacker created a worm which posted itself on the users timeline when the users saw and clicked the malicious tweet in their feed.

Related evaluation points:

- *Cross-site request forgery (CSRF)*

Links:

- CSRF attack strikes Twitter
- Don't Click The WTF Link On Twitter Unless You DO Like Sex With Goats

Veeder-Root

Date: 2015-01-23

Gas stations use automated tank gauges (ATGs) for remote control and diagnostics. Automated tank gauges were exposed to Internet through serial port servers that map ATG serial interfaces to the Internet-accessible TCP port.

Most of ATGs were manufactured by Veeder-Root, a petroleum equipment service company. The system allows maximum of six letters password, but often the password is not set.

The attacker could change the calibration and make the tank report invalid full or empty status. In the worst case, the attacker could shut down the pumps of a gas station.

The hackers laters exploited this and toyed with US gas stations.

Related evaluation points:

- *Non-public administration site*

Links:

- Internet attack could shut down US gas stations
- Thousands of U.S. gas stations exposed to Internet attacks
- Mideast Hackers May Be Attacking US Gas Stations

XCode

Date: 2015-09-17

XCode is Apple's development tool for building iOS and OSX applications. A pirated version was distributed with an ability to infect all created applications with malware code.

A pirated version of XCode was popular among Chinese iOS developers. The development tool is large (3GB) and downloading it from official Apple sources is very slow in China.

Chinese developers used the pirated XCode to create applications, leading to compromise of many official Chinese applications in App Store. The high value targets included an official application from Baidu, a large Chinese search engine.

Apple's App Store review policies did not caught the malware and rigged applications passed the review.

Later Apple made a Chinese mirror for XCode downloads.

Related evaluation points:

- *Software installation from safe sources*

Links:

- Novel Malware XcodeGhost Modifies Xcode, Infects Apple iOS Apps and Hits App Store (PaloAlto Networks)
- Apple will host Xcode on Chinese servers following malware attack

2.12 Contact

For private issues related to your service, its security and if it possibly has been compromised please contact mikko@opensourcehacker.com.

For spelling fixes and updates find the appropriate source files on Github and use Github's Edit button.

For questions and clarifications regarding the guide content open an issue on Github.

2.13 License and copyright

The material is licensed under Creative Commons Attribution 4.0 International (CC BY 4.0) license.

You are free to: