Chapters

# Chapter: team

|  |
| --- |
| Basic security practices |
| The team members are trained and follow the basic security practices |
| The team members are trained and aware of common cyber threads like phishing attacks, social engineering. They can identify the basic attacks, like spearhead phishing emails, and do not fall victim to those. The team members maintain cyber hygiene and do not use working devices to visit sites which might compromise the web browser. Software, both desktop and server software, is maintained in up-to-date versions where know vulnerabilities do not exist. |

|  |
| --- |
| Unsafe file attachments |
| Potentially dangerous file attachments are handled securely |
| File attachments in email and chat are one of the most common attack vectors. is The desktop applications opening likely rigged payloads, like office suite files, PDFs and Flash animations are disabled or the team members do not use them to view the files. Instead, the suspicious attachments, especially ones coming outside the security barrier, are opened in a web browser based viewer or similarly sandboxed tool. |

|  |
| --- |
| Password manager |
| The team members use password manager |
| All team members use password manager for their passwords. The password manager is the only cognitive sane way to manage a lot of sensitive, strong and random passwords. Without randomized passwords, a compromised third party site may lead to loss of other passwords due to password reuse or password pattern reuse. Whether one can trust third party service to store password is a subject to discussion depending on the sensitivity of the project. |

|  |
| --- |
| Third party devices |
| The team members do not use third party devices for logging in |
| If the device comes from an untrusted party, it may contain keyloggers and other malware to record the user actions. Such devices include internet kiosks and other free terminals. The team members use only their own working devices for the work. |

|  |
| --- |
| Encrypted computers |
| The team members have disk encryption enabled on their personal computers |
| The permanent storage, SSD or hard disk, on the team laptops and desktop computers is encrypted.  This implies the usage of disk encryption technology like FileVault (OSX), dm-crypt (Linux) or BitLocker (Windows).  A lost device, when encrypted, cannot lead to any kind of compromises as password is always required to unlock the device. Even if the device wakes up from hibernation. |

|  |
| --- |
| Encrypted mobile devices |
| The team members have disk encryption enabled on their mobiles and tablets. The devices are password protected |
| A lost device, when encrypted, cannot lead to any kind of compromises. Even if the device were not to contain sensitive data it could contain active email inboxes and team chats leading to further account compromise and phishing. |

|  |
| --- |
| Two-factor authentication on email |
| The team member email accounts require two-factor authentication to log in |
| Email accounts contain sensitive information and they can be used to reset the master password of services and infrastructure. Email account is also attractive target to hack in as they are either public or easily guessable. Even if email account is protected by 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. |

|  |
| --- |
| Two-factor authentication on critical services |
| Administrating infrastructure services requires two-factor authentication |
| The team relies on third party services for infrastructure: hosting, domain name, certificates, email, SMS, attack mitigation proxies, etc. If these services provide a two-factor authentication this option is used. This adds additional layer of security if the infrastructure provider becomes a target of attack and the master password can be reset e.g. through phishing. |

|  |
| --- |
| Two-factor authentication on the administrative site |
| The administrative part of the website requires two-factor authentication |
| Usually the Internet services provide an administrative site or a backend site where the site managers and support personell can perform in-house tasks. If the attackers compromise the passwords of the team members they should not be able to get in to the administrative site just with the password. Instead, a two-factor authentication token is required for the site admins to log in. Furthermore the administrative address can be limited to VPN or other well-known (office) IPs. |

|  |
| --- |
| Passphrase on server login keys |
| The server logging in is by keys only which are passphrase protected |
| The logging in to production or staging servers is only allowed with the key files. The key files are passphrase protected. The usual logging method is by SSH, but if alternative methods exist accessing the servers they must provide similar method. This protects against brute force attacks against devop access. Furthermore keys must be passphrase protected so in the the case keys are accidentally leaked they are not useful. |

|  |
| --- |
| Two-factor authentication on server login |
| The server logging in requires one time token |
| The server login is further restricted to two-factor authentication, so that even in the case the devop laptop is hijacked by malware, this laptop cannot login to the server without a token from an external device. |

|  |
| --- |
| Audited server login keys |
| A real-time method of maintaining and revoking keys across all servers is used |
| In any point of time, the system administrators of the company can revoke any key in the whole organization. Full audit logs of key usage is available and stored separately. This allows quickly to address issues when a key compromise is suspected. |

|  |
| --- |
| Software comes from secure sources |
| Software installation comes from knonw good sources |
| Pirated software is riddled with malware. The team installs software which comes from legit sources only, reducing the risk the software comes with maware. |

|  |
| --- |
| Sensitive data access limitations |
| Backend sensitive data access is limited |
| If multiple people access the backend data, the access is limited in a way that the sensitive information is not exposed unless necessary for performing the work. |

|  |
| --- |
| Sensitive data access logs |
| Backend sensitive data access is logged |
| All actions of team and support persons accessing and manipulating sensitive data are logged. In the case of privacy breach claims these logs can be used to reconstruct the scenario who have been accessing the data. See also :ref:`log-server`. |

|  |
| --- |
| Data scrubbing |
| When working with datasets, it is cleaned from sensitive information |
| Instead of working with full datasets, there exist a repeatable process of making a cleaned dataset with reduced sensitive information from the production data. This cleaned dataset is given for the team members who need to analyse, test and develop against the data. This limits the impact of data dump leak in the case the data dump ends up to the hands of an unknown party. |

# Chapter: web

|  |
| --- |
| HTTPS / TLS only |
| The service is HTTPS only with security HTTP headers |
| The service offers traffic only over encrypted channels. Any sensitive service has no excuse not to to use full encryption anymore. It is well known that powerful actors tap and modify Internet traffic globally. Furthermore, the HTTP response headers should include security headers, like HTTP Strict Transport Security and X-Frame-Option which mitigate the man-in-the-middle attack risk in the case the user's network is compromised. |

|  |
| --- |
| Database injection |
| The software is written in a manner that there is no possibility of database injection attack |
| One of the most common web application vulnerabilities is a database injection attack. In the most cases, the database is SQL based, providing opportunity for SQL injections. This can be easily prevented by never constructing database statements by hand and always using a framework to construct the queries, so that all values are properly escaped. The manual SQL manipulation should be prevented from the application developers, so that no room is left for a human error. |

|  |
| --- |
| Cross-site scripting (XSS) |
| The software is written in a manner that there is no possibility of cross-site scripting attack |
| Cross-site scripting attack is a way to perform actions on the behalf of the user when the user views or clicks a compromised payload. The attack target can be the site visitor or the site administrator. The usual cross-site scripting attack is posting text or file attachment which payload is not well-escaped HTML. This attack can be avoided by using a proper software development framework which always escapes variables in output and not relying the developers to manually escape variables in page templates, JavaScript or HTML JSON embeds. |

|  |
| --- |
| Cross-site request forgery (CSRF) |
| The software is written in a manner that there is no possibility of cross-site request forgery attack |
| Cross-site request forgery is an attack where a JavaScript payload hosted on a third party site performs attack on the behalf on the user on the targeted website. The software should be written using a framework which prevents HTTP POST submissions without CSRF token. Any state changing action (login, create, modify, delete) should not be GET request. |

|  |
| --- |
| Password storage best practices |
| The user passwords and two-factor seeds are hashed and salted so that bruteforcing them is not viable. |
| This protects the user password integrity in the case the database is compromised. The developers should not do password management themselves, but use a specialized library to do the password hashing and salting for persistent storage. |

|  |
| --- |
| Non-public administration site |
| The administration site is not easily accessible to public |
| Many common software platforms come with the default administration site in a location like \*/admin/\*. If the administrative URLs are well-known the attacker can exploit this and guess weak administration interface HTTP endpoints to exploit those. The administration interface should be in non-guessable, non-end user visible, URL. Furthermore the additional access restrictions can be placed with two-factor authentication, VPN and IP restrictions (see :doc:`Team security <../team/index>`). |

# Chapter: user

|  |
| --- |
| Two-factor authentication |
| The site users are encouraged to use two-factor authentication |
| The site user loes their password through many channels: passwords are recycled across multiple sites (e.g. email account), passwords are too weak, password are given out through phishing emails or devices are compromised by malware. In this case, the two-factor authentication should stop the attacker with a mere password to access the user account.  Having the two-factor authentication as optional is not often enough, as the users only see the reduced usability (longer login process) and are not aware of thread models. Incentives, like reduced fees, should be used to encourage enabling the two-factor authentication. From business perspective, this can be justified as reduced support cost of dealing with hacked account cases.  The most sensitive operations, like where money is transferred out from the system, should require minimum of two different two-factor tokens: one for login and one for transfer. This makes two-factor intercepting phishing site operation less robust, as the users are more likely notice bad URLs if they need to spend more time on the phishing site. With only one authentication token the phishing site can do transfer out on the second the user hits the login, making phishing more likely to success.  Popular two-factor authentication methods include Time-Based One Time Password (TOTP, mobile apps, Google Authenticator), One time pad (HOTP, paper codes, used by many European banks), SMS and hardware devices (like YubiKey).  External services like Authy and Clef provide two-factor-as-a-service.  Google Authenticator is a popular open mobile two-factor app. Despite the name says Google, you can use it on your own site, the application can be used independently of Google services, offline, the RFC is open and there are multiple open source implementations.  .. note::   SMS is not deemed secure in large scale. SMS messages are intercepted by mobile malware. SMS may travel in plain text and various parties in operator business can read them. Mobile number portability opens a vector for the attacker to gain access to the victims phone number. SMS may not be reliable in third world countries, thus making it not viable option for global business. |

|  |
| --- |
| Third-factor authentication |
| The user login goes through additional check in abnormal circumstanses |
| The users might not have two-factor authentication enabled. Even with the two-factor authentication there is a chance that the tokens are compromised through a phishing site. In these cases the service should detect abnormal conditions and perform additional checks before letting the login to proceed.  The common third factor authentication criteria include \* Country of the IP \* New device or browser by stored permacookie  In these cases the service should prompt the login to go through additional verification step. This could be  \* Email confirmation \* SMS confirmation  .. note ::   Third-factor authentication does not protect against cases where the device of the user is compromised by malware and the service cannot differetiate between the legit and the malicious traffic coming from the same device. |

|  |
| --- |
| Brute force login prevention |
| Site login attemps are throttled in multiple ways |
| The attackers try to brute force the logins of the users. The site should take adequate measures that so that multiple login attempts are effectively stopped.  There are few different password brute force attack modes:   \* Spearhead bruteforcing targetting a single user   \* Email and password combination guessing from a third party site leak or blackmarket   \* Email and common password list guessing, like 1000 most common passwords   \* Scraping the site for user account names and then combining them with above  The attacker may be in possession of thousands of IP addresses.  The counter actions should include:   \* CAPTCA on second login (allow one wrong password attempt per user)   \* Prevent login attempts per IP (fail2ban)   \* Prevent login attempts per username (spreadhead attack)   \* Force all users to go through CAPTCHA before login if the system global login rate is abnormal high (botnet-based attack)  Relying solely to CAPTCHA to prevent brute forcing is not recommended, as the automated CAPTCHA solving success rates are counted in tens of percents.  Beside the security ramifications, 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 the 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 to increase the account security. |

|  |
| --- |
| Effective session kill |
| When the user account is deactivated, all related sessions are killed |
| If the attacker gains access to an user account the system administrators must be able to kick out the attacker. The account deactivation may only affect the database records of the account, not dropping the active HTTP sessions which are stored in a separate store. When an user account is deactivated, all communication channels to this user must be dropped.  All user sessions should be dropped on  \* Account delete  \* Password change  \* Email change  \* Third factor authentication |

|  |
| --- |
| User audit logs |
| The service keeps audit logs of sensitive user actions |
| All sensitive actions of the users should be logged to a user specific action list. In the case case of a crime, the user audit log may be handed to the officials. The user itself may or may not review his past actions based on this list.  The list is also important to protect the service operator itself against fraud. For example. the user can arrange stealing of the user account. The thief transfers the assets of the user to the friendly party of theirs. Then the user can blackmail and threat to sue the service unless the user is (incorrecly) reimbursed. The user audit logs prove the correct password and authentications codes were used to initiate the transfer and shift the resposibility to the users themselves.  The log should include at least:  \* The 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   Furthermore the user audit logs can be used to recover the system in the case of flaw leading to a mass account compromise. |

|  |
| --- |
| Account verification process |
| The creation of bogus accounts is prevented |
| This only applies for services where users can interact with other users or the world e.g. spam and harrash them.  To keep the system clean, one should prevent the creation of fake and robot accounts. The cost of automatic account creation should be so high that there is no financial gain to use the account for automated harrashment. The account creation proces should be still easy enough not to discourage the users to sign up.  The account verification is also important for anti-money laundering (AML) and know-your-customer (KYC) cases where it is imperative to know one is dealing with the rightful holder of the financial assets.  The 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. |

# Chapter: infrastructure

|  |
| --- |
| Login throttle |
| Throttle or ban IPs with multiple login attempts |
| Prevent bruteforce attacks against your service by blocking multiple login attempts. asdf 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 is not effective against adversaries with botnets with thousands of global IPs in their posession. |

|  |
| --- |
| Security proxy |
| Servers are behind security proxy |
| Using a security proxy service hides the IP of your servers from the attacker, thus making denial-of-service attacks more difficult to perform. The security proxy services are provided by specialized companies who possess geographically distributed servers, a lot of bandwidth and can thus withstand attacks. Furthermore these services apply rules on the traffic to prevent common malicious visitors and crawlers. |

|  |
| --- |
| Internal domains not exposed |
| Internal services cannot be discovered from public data |
| Internal services, testing and staging servers, which are likely to be less protected and monitor, should not be publicly known. The public DNS records should not expose any services which are not public. For internal services run a custom DNS server or use a non-guessable secondary domain name. |

|  |
| --- |
| Log server |
| Critical logs are mirrored to a append only 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 server or service should have different access credentials from the administrators of normal systems. In the case the attacker gains access to infrastructure, this prevents wiping or manipulating logs. This allows robust recovery and post-mortem from potential infrastructure attack. |

|  |
| --- |
| Encrypted server data |
| Data is stored on encrypted partitions |
| All user data should be stored on encrypted partitions or files. In the case of unauthorized physical access the data cannot be compromised. Encryption should apply for backups and other offsite files too.  Disk encryption protects the 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 a passphrase to unecrypt the data partitions. The easiest way to achieve this is to have separate partitions for boot volume and data volume. Having separate "high" and "low" states the server cannot enter to to the state with access to data unless an authorized person enters a passphrase through a terminal.  This protects the loss of data against  \* Hosting provider attacks or social engineering attacks against the hosting provider  \* Law enforcement attacks  .. note ::   All virtual machines, like ones provided by Amazon EC2, 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 your adversiers include law enforcement agencies and nation state actors it is recommended to use only physical servers which cannot be rebooted without notice. The servers should be enabled with physical chassis removal detection which stop at BIOS boot if they detect the server chassis to be opened.  Furthermore all dedicated servers which you do not install yourself in your own rack can be deemed as unsafe. If you do not have the key to th physical lock at the rack you can never be sure that somebody tampers with the server. |

|  |
| --- |
| Secure server-to-server connections |
| Server-to-serve 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 that anyone tapping a physical cable cannot any extract any information.  The connection encryption methods include SSH tunnels and VPN. |

|  |
| --- |
| Publicly exposed services and firewalling |
| Unnecessary services are not exposed to Internet |
| All private services like databases, queue services and caches should be not Internet accessible.  The services should bind to private network or localhost IPs only.  The easiest method to verify this is to scan the ports of all public IPs. Only the publicly accessible endpoints, like HTTP and HTTPS, should be available.  .. note ::   A firewall should be only a secondary measure. By default the services should be configure in a manner that they to do not bind to publicly exposed IPs. Furthermore firewalling outgoing connections might be problematic, as many services rely on third party API service today. |

|  |
| --- |
| Intrusion detection |
| Intrusion detection alerts on unexpected server activity |
| Intrusion detection software monitors the servers and alerts in the case there is unexpected activity.  Intrusion detection is a measure to detect compromised servers. Intrusion detection software monitors processes, file system, configuration files, passwords and user database. In the case there are changes not matching the predefined ruleset an alert is fired. |

|  |
| --- |
| Priviledge separated software installation |
| Software installations are under custom accounts and rights |
| Any installed software is under non-root (non-admin) account. The compromise of the software throguh an exploit cannot compromise the server as a whole. |

# Chapter: assets

|  |
| --- |
| Cold wallet |
| Cold wallet maintains the most of assets offline |
| In the case of the compromised service, the attacker cannot get access to all assets and only can steal minor part of them, severely limiting the damage. Most of the assets are stored offline, in a non-connected computer, which has only physical offline access in a safe location. Thus, the attack taking all the assets would need a physical access to this device, making over-the-internet attacks impossible. |

|  |
| --- |
| Transaction verification |
| Outgoing transactions are verified by heurestics |
| Outgoing transactions are verified by heurestics, so that unusual transactions need manual verification or other human interaction. This prevents emptying the hot wallet in the case private keys or hot wallet API access is compromised. TODO |

|  |
| --- |
| Multisignature for major transactions |
| Minimum of two parties are required to make major a customer assets transfer |
| A sole person alone should not be able to embezzle the customer assets.  Digital currencies provide a multisignature mechanisms needing the approvement of at least two different parties to sign the transfer of the assets. Such a mechanism should be used any time a large fraction of customer assets are moved e.g. topping up the hot wallet from the cold wallet. This decreases the risk of corruption or blackmail attacks. |

|  |
| --- |
| Race condition prevention |
| A systematic development method prevents race conditions |
| A systematic development method is applied to all transactions, so that race conditions cannot exist in financial transactions. Otherwise exploiting a 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 applie |

Incidences

|  |
| --- |
| bitstamp |
| Bitstamp |
| Bitstamp 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 private and confidental memo by the general counsel of Bitstamp, leaked what happened. Though the authenticity of the memo is not confirmed by the authors, it is believed to be genuine.  "On 4 November 2014, Mr Merlak [CTO of Bittamp] 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 software and infrastructure security procedures, they left team members exposed. By using vulnerable production software suite to view the document instead of a web based viewer, the laptop of a high valued target was compromised. Even though the server required two-factor login, because two-factor was disabled in certain circumstances for the working convenience it didn't stop the attackers this time. Furthermore, during the incident, their Bitcoin transaction system did not use any kind of fraudulent transaction detection mechanism which could have stopped the attacker. |

|  |
| --- |
| ashley-madison |
| Ashley Madison |
| Ashley Madison, a service billed as enabling extramarital affairs, got comprehensibly compromised. A Canadian, Avid Life Media, was running a dating site for married people.  "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 hacking happened. It is known that a blackhat 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 internal network.  The CEO of Avid Life Media claims the breach was by an insider who was not an employee. |

|  |
| --- |
| bitly |
| Bitly |
| Bitly unecrypted backups got compromised.  Bitly is a URL shortening service. The users can log in with their Facebook and Twitter accounts. In the incidence, the attacked 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)  The authoritative report "More detail", by Bitly, is now taken down (http://blog.bitly.com/#85169217199). |

|  |
| --- |
| cloudflare |
| CloudFlare |
| Matthew Prince, the CEO of CloudFlare, a security proxy service company, 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 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.  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 feature was mostly 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. |

|  |
| --- |
| xcode |
| XCode |
| XCode is Apple's development tool for building iOS and OSX applications. A pirated version was distributed with an ability to infect all applications created with the pirated versions. Many official Chinese applications in App Store got rigged. The high valued targets included the official application of Baidu, a large Chinese search engine.  Apple's App Store review policies did not caught the malware and rigged applications passed the review.  The reason why Chinese developers used the pirated XCode in the first place is that the development tool is large (3GB) and downloading it from official Apple sources takes forever in China. |

|  |
| --- |
| slack |
| Slack |
| Slack is a popular team communication tool among software companies and in US. The database of Slack got compromised, leading to the exposure of salted passwords.  After the breach Slack detected suspicious activity targetting 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. |

|  |
| --- |
| lastpass |
| LastPass |
| A popular password management service, LastPass, got compromised.  LastPass account email addresses, password reminders, server per user salts, and authentication hashes were compromised.  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. |

|  |
| --- |
| chinese-android |
| Asian Android phones |
| Various (low budget) Asian Android phones 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. |

|  |
| --- |
| nasa |
| NASA |
| 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. |

|  |
| --- |
| tor |
| Tor |
| 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\* 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 recorded cases where the victim accepted the invalid HTTPS certificate even though Firefox-based Tor browser presented a red warning screen with difficult options to proceed beyond it. |

|  |
| --- |
| soho |
| Soho |
| 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. |

|  |
| --- |
| twitter |
| Twitter |
| Twitter allowed to post a tweet using a HTTP GET request.  The attacker created a worm which posted itself on behalf of the user when the users clicked a link they saw in their friends feed. |

|  |
| --- |
| sebastian |
| Sebastian |
| A hacker group TeamBerserk claimed to have stolen more than 100k USD via SQL injeciton injection from Sebastian, a Californian based ISP.  Through SQL injection, the attackers downloaded the list of ISP's customers, their usernames and passwords in clear text. The attackers exploited the fact the users recycle the same password and used usernames and passwords login GMail, PayPal, CitiBank, etc. The attack was demostrated on a video uploaded to MEGA (now defunct).  Tom Dominico, 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.” |

|  |
| --- |
| facebook |
| Facebook |
| Facebook status update functionality did not properly escape parameters.  It was possible to post HTML content which was not properly sanitized which further loaded JavaScript. The loaded JavaScript then took actions on the behalf of the user. This allowed the attacker to create a worm which propagated through Facebook walls.  The root cause was 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. |

|  |
| --- |
| veeder-root |
| Veeder-Root |
| 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 bt 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 full or empty. Worst case the attacker could shut down the pumps. |

|  |
| --- |
| icloud |
| Apple iCloud |
| Apple iCloud service was subject to login brute force attack leading to the compromise of several celebrity accounts.  Apple did not follow the security best practices to prevent brute forced login attempts. Find my iPhone, a part of iCloud services, allowed unthrottled login attempts.  Later the private photos of victims, most of them being celebrities, were leaked in Internet, causing harm to these people.  Apple did not apologize. |

|  |
| --- |
| sms-malware |
| SMS intercepting trojans |
| Multiple malware and trojan programs are observed to steal SMS two-factor authentication codes. These mostly target banks and popular services. Malware is mostly Android ecosystem issue, though other operating systems, especially jailbroken ones, can be infected.  When the user receives two-factor authentication codes over SMS they are forwared to the attacker. Furthermore the malware intercepts logins and passwords to popular services. |

|  |
| --- |
| instagram |
| Instagram |
| Instagram deleted millions of accounts.  Due to lax account creation process, A large proportion of Instagram accounts were fake and automatically created robot accounts. The fake accounts can be exploited as fake followers or to send spam. Social media PR companies may buy fake followers to inflate their campaign success rates.  It can be speculated that even if being aware of the severity of the issue Instagram was not in rush to delete the fake accounts before acquisition by Facebook to inflate their market value.  Some celebrities lost up to 90% of their followers. Instagram's own Instagram account lost 30% of its followers. |

|  |
| --- |
| steam |
| Steam |
| A flaw in password reset procedure allowed login to any Steam account without two-factor authentication.  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 to the email. User accounts with two-factor authentication enabled were protected.  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. |

|  |
| --- |
| hacking-team |
| Hacking Team |
| Hacking Team was a company selling offensive intrusion and surveillance capabilities to governments. Hacking Team got compromised, all 400GB of internal data leaked.  All the stolen information was likely accessed via the compromised computers of Christian Pozzi and Mauro Romeo, two Hacking Team’s sysadmins.  The leaked data demostrated Hacking Team operations security standards were not high. 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 is still not known. |

|  |
| --- |
| cryptoine |
| Cryptoine |
| A race condition existed in the software of Cryptoine, now defunct Bitcoin exchange.  The race condition allowed the attacker to drain all hot wallets.  This damage caused the closure of the exchange. |