ICP-3011 Computer & Network Security

Reflect And Relate

Max Petts - *eeub35*

PRIFYSGOL BANGOR UNIVERSITY

17 - **04.10.2020**

Week 1 - SSH Remote Access

Concept learnt:

- Authentication

In order to permit or deny access to the rightly entitled resources, the security system must be able to differentiate between users; this is identification. When discussing authentication the user being identified is called the Principal. Authentication is the act of validating the claimed identity, and so contains two parts: identification and verification.

Authentication is apparent in many day to day tasks for example, when you get tested for Covid-19 the nurses will ask you to recite your name and address, as to verify you're the person you claim to be. Your name and address is called a credential and is used to verify the principal being claimed is entitled to. Without this step Principal may get their test back positive, when it is actually negative, or vice versa. Some methods of authorization are less user friendly and become laborious, but these credentials are the most secure, as they are harder to brute force due to them using something the principal owns, or the principal themselves (retina scans).

I use git to track changes to my projects which authenticates the user with a public key, instead of inputting my username and password, before any changes can be pushed to the repository; if this repo contained sensitive information then the key should use ECDSA, as RSA is vulnerable to timing attacks[1].

Week 2 - GPG - GNU Privacy Guard

Concept learnt:

- The CIA (Confidentiality Integrity Availability) Triad

The CIA triad is described by f5.com as "so foundational to information security that

anytime data is leaked ... you can be certain that one or more of these principles has been violated."[2]. The center point of this model is the data being stored, whereas other models focus on Identity and Access Management(IAM), permissions management and data classification. A multitude of security models should be used so defence in depth can be achieved, instead of just the CIA triad.

Confidentiality ensures that data "is not made available or disclosed to unauthorized individuals, entities, or processes."[3]. An example of a breach in confidentiality, that has happened to me, would be a new employee using a current employees login to do their training and viewing a confidential file already open.

Integrity is defined in the oxford dictionary as "the state of being whole and not divided"[4], but in the InfoSec(Information Security) realm it refers to the reliability, authenticity and correctness of the stored data. The Principal is usually the one responsible with keeping all information up to date. This relates to the new GDPR regulations, "Article 5(1)(f) of the GDPR concerns the ‘integrity and confidentiality’ of personal data"[5]. The CIA triad is also heavily referenced throughout the document, affirming its importance.

Availability mostly refers to the hardware aspects of security systems, as ensuring the system remains powered on and connected to the internet is vital, no matter the literal application. A few examples of breaches in availability are: power outages and DDoS(Distributed Denial of Service) attacks. Power outages may be circumvented through the use of a UPS(Uninterruptible Power Supplies), whereas preventing DDoS attacks are harder than buying a new piece of equipment.

Week 3 - E-Mail Spoofing

Concepts learnt:

- Security from the start

When developing a piece of software, security should always be the first port of call. If the security of a system is left as an after thought then the trust is compromised. This concept is similar to the Layered Defence Model in regards to the need for lower layers of security systems to be enforced with appropriate security protocols - if a lower layer fails every layer above it also does.

A prime example of not baking security in from the start would be zero-day vulnerabilities. This is defined by Norton as "a software security flaw that is known to the software vendor but doesn’t have a patch in place to fix the flaw."[6]. Zero days are usually vulnerabilities that have been present in the software since its conception.

Recently a new zero-day has been found within all chromium based browsers. The vulnerability itself exists within the FreeType software library, used to render fonts, and is a memory-corruption flaw which causes a heap buffer overflow[7]. It was reported by Sergei Glazunov of Google Project Zero on 19/10/2020, and affects most operating

systems, from Windows, Linux Mac and Chrome OS. Unfortunately Google has removed the official report of this[8]; probably akin to the severity of the exploit within a widely used font rendering library.

In the future when developing software, instead of creating new functionality as quick as possible, I will think about the security behind features first. As building security on top of previous insecure features lacking in security layers means nothing if not secure, so the security of the software should be perfected before new features are added.

Week 4 – 'Real World' Cracking

Concepts learnt:

- Validation

Whenever the end user has the ability to input data it should be pruned and checked, even if this data is not going to be directly ran, or inserted into a database. Validation is the process of ensuring data is complete, accurate and secure; informing a user if their input does not meet the expected structure (e.g inputting a name in the DoB field). If validation is overlooked it is mostly associated with an overwhelming confidence in the end users capability of correct data input. It should be used to not only prevent malicious attacks but also incorrect data - it is responsibility of both the back and front end engineers.

W3schools claims "SQL injection is one of the most common web hacking techniques."[9], and can enable malicious code to be ran against the database. However, this can be all be avoided by, you guessed it, validating the users input. In PHP, this is done using the **filter_var(*variable*, FILTER_SANITIZE_SPECIAL_CHARS)** function which strips '"<>& and any other characters with an ASCII value lower than 32[10].

I believe this lab has not altered my views on this concept as, in my opinion, it is best practice to treat everyone like they are uneducated on the subject until proven otherwise. However it is also necessary to accompany this approach with a juxtaposing outlook that there will always be someone smarter than you; who is able to break your validation and security measures. As so every opportunity should be taken to validate input and subsequently secure your software.

References

[1] http://crypto.stanford.edu/~dabo/papers/ssl-timing.pdf

[2] https://www.f5.com/labs/articles/education/what-is-the-cia-triad#content

[3] https://books.google.co.uk/books?id=DvdICAAAQBAJ&lpg=PA100&pg=PA100#v=onepage&q&f=false

[4] https://www.oxfordlearnersdictionaries.com/definition/english/integrity?q=integrity#integrity_sng_2

References

[5] https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/711097/guide-to-the-general-data-protection-regulation-gdpr-1-0.pdf

[6] https://us.norton.com/internetsecurity-emerging-threats-how-do-zero-day-vulnerabilities-work-30sectech.html

[7] https://www.welivesecurity.com/2020/10/21/google-patches-zero-day-flaw-chrome/

[8] https://chromereleases.googleblog.com/2020/10/stable-channel-update-for-desktop_20.html

[9] https://www.w3schools.com/sql/sql_injection.asp

[10] https://www.php.net/manual/en/filter.filters.sanitize.php