TRUSTED TECH TALKS

WEBINARS · EVENTS · NETWORKING

Encryption Isn't Magic: Hackers Can Break It

Jake Adshead
Cyber Security Consultant
maxwellbond

Holly Grace Williams
Managing Director
secarma
CYBERSECURITY EXPERTS

Introduction

The sophistication and complexity of cyber-attacks continues to increase meaning that now more than ever, businesses need improved and effective cyber security to reduce risks and protect their data and information. However, businesses often suffer from skill and knowledge gaps in cyber security, especially start-ups and SME's who might not have the luxury of dedicated cybers security teams. That's why we invited Holly Grace Williams, Managing Director of Secarma, the cyber security consultancy, to join us on our recent Trusted Tech Talk to discuss all things cyber security, risks, and tooling. You can re-watch the full webinar on our YouTube channel or read the highlights below.

Security Testing Overivew

Vulnerability scanning is a fully automated approach to security that performs security testing on a defined scope, usually using a software platform. It's a cost-effective way to perform very basic security testing to identify common misconfigurations and missing patches. However, other than this it is relatively unintelligent.

Penetration testing is a human led, scope-restricted security test of a system intending to exploit vulnerabilities to determine the real-world risk of system security. Essentially penetration testers try to identify as many vulnerabilities as possible to then prioritise changes and action points within a security report, the final part of which is usually focused on system hardening.

System Hardening is a collection of tools, techniques, and best practices to reduce vulnerabilities in tech applications. It is a critical part of cyber security, but it is often something that is put at the back of a report because it is not necessarily attached to one specific vulnerability. When approaching penetration testing, businesses will work through the critical issues until they run out of time so more often than not, they never get to that part because they simply don't have the spare resources. By not actioning system hardening suggestions, businesses miss out on the chance to reduce future risks because the more system hardening you do, the harder exploitation is. It can slow a hacker down, therefore making response and remediation easier and quicker.

When businesses naturally prioritise single vulnerabilities, it means that system hardening often becomes secondary. This is problematic because system hardening will actually make cyber attacks and exploitation much more difficult, if not stop it all together, as it can slow attackers down and give you more time to intervene, respond and remediate before any information, systems, or data is compromised. Therefore, it is fundamental to improving cyber security, but is often overlooked.

Cryptography

What does 'broken' mean?

When an aspect of the cryptography is ‘broken’ it could mean that it is academically broken, or it could be practically broken. But what is the difference between academically broken and practically broken? If cryptography is described as academically broken it means that a known weakness exists. If cryptography is described as practically broken it means that a fast, reliable, and publicly available exploit exists.

HTTPS, TSL, and HSTS: Security & Connection

When we connect to websites and other technologies most people will immediately check to see if the padlock is in the URL bar and to check the site is HTTPS, incorrectly thinking that this demonstrates that the site is completely secure and safe. This, however, is not the case. The padlock only tells us that the web browser hasn’t detected that the connection to the web server has been modified in anyway. It simply tells you the connection is good, not that a website is secure or trustworthy, or validated.

TLS provides security of data in transit and supersedes Secure Sockets Layer (SSL). It is a protocol that controls your connection to a server and allows you to utilise encryption in a simple way to ensure data is protected when in transit.

Although TLS superseded SSL, there are several versions of each (SSLv2.0; SSLv3.0; TLSv1.0; TLSv.1.1; TLSv1.2; TLSv1.3) and the basic advice here is to use the most up to date version, in this case it is TLSv1.2 or TLSv1.3. The biggest challenge to this is often compatibility. When making these changes it’s important to consider how the change will affect the user base. However, in terms of TLSv1.2, Google Chrome has supported this by default since 2013, so there really is no reason why you need to be using or supporting browsers over 6 years old. Additionally, you should also be looking at X509 certificates and ciphers.

There is a problem with TLS. Whilst our data is in transit across the web, it's in an encrypted format so that no one can read it or tamper with it. The browser and server decrypt the data at each end so they can process it and act on it. The browser decrypts the data and renders a webpage, and the server might decrypt the data and receive a new request or verify your login credentials. If you're signing up to a new website that uses HTTPS, as all of your details traverse the web, they're encrypted. This prevents anyone with access to that connection from stealing your private and sensitive data. As the server receives your data, it decrypts it and then most likely stores it in a database so it can remember who you are in the future. Unfortunately, TLS doesn't give us any assurances about how or where that data is stored.

When you first connect to a website, it is likely that you will connect using HTTP (insecure), the web application will then request that you connect securely. These messages will be in plain text, which can obviously be played with and modified by threat actors during an interception attacks. These interception attacks, also known as ‘man in the middle’ attacks, can happen quite frequently.

Whilst that initial connection is insecure and in plain text, there is a tool to fix this. HTTP Strict Transport Security (also named HSTS) is an opt-in security enhancement that is specified by a web application through the use of a special response header. Once a supported browser receives this header that browser will prevent any communications from being sent over HTTP to the specified domain and will instead send all communications over HTTPS. It also prevents HTTPS click through prompts on browsers. HSTS automatically redirects HTTP requests to HTTPS for the target domain and does not allow a user to override the invalid certificate message.

Ciphers: RC4NOMORE

HTTPS supports several encryption techniques and ciphers, one of them being RC4. RC4 was previously used 50% of the time, but the RC4NOMORE attack exposed weaknesses in this RC4 encryption algorithm. More precisely, in most situations where RC4 is used, these weaknesses can be used to reveal information which was previously thought to be safely encrypted.

An attacker would be able to decrypt web cookies, which are normally protected by the HTTPS protocol. Websites use these cookies to identify users and authorize actions they perform. By obtaining the cookie of a victim, an attacker can log into a website as if they were the victim. This means the attacker can perform actions under the victim's name. The estimate was that it would take a hacker 75 hours to decrypt a cookie, although in a real life test, it took only 52 hours to complete the attack.

Avoid ciphers and cipher modes with known vulnerabilities.

Cross-Site Scripting (XSS) and Reflected XSS

XSS is often reported as the most common web application vulnerability and is in the top ten vulnerabilities to care about in the Open Web Applications Project and occurs where user supplied input is insecurely included within a server response. Exploitation of this vulnerability can result in virtual defacement, data theft, privilege escalation, or malicious software distribution.

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

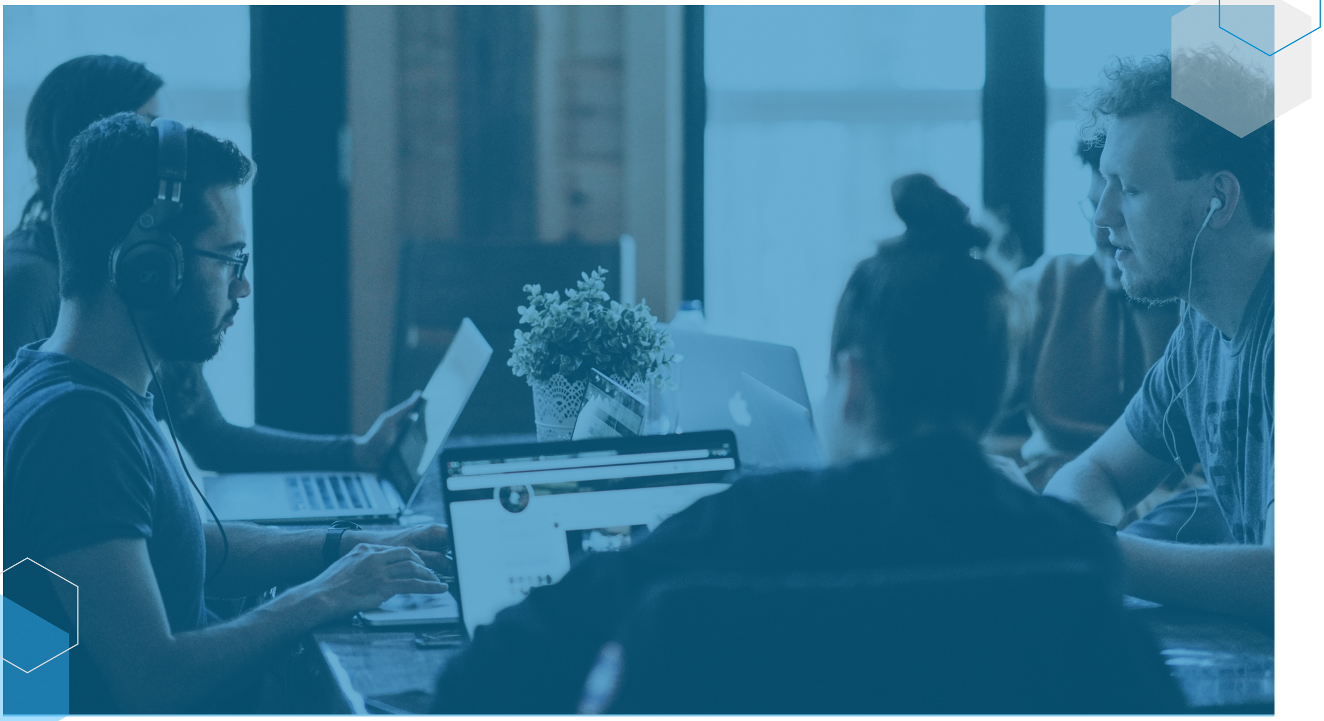
An attacker can use XSS to send a malicious script to an unsuspecting user. The end user’s browser has no way to know that the script should not be trusted and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

Reflected XSS the user supplied information is part of the URL. Reflected attacks are those where the injected script is reflected off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. Reflected attacks are delivered to victims via another route, such as in an e-mail message, or on some other website. When a user is tricked into clicking on a malicious link, submitting a specially crafted form, or even just browsing to a malicious site, the injected code travels to the vulnerable web site, which reflects the attack back to the user's browser. The browser then executes the code because it came from a "trusted" server. Reflected XSS is also sometimes referred to as Non-Persistent or Type-II XSS.

A basic example would be if a threat actor sends a crafted link to the target user via email. If the target user clicks then that link, a payload is executed on the server. This allows for things like virtual defacement as mentioned before. The link doesn't have to be sent to a user directly either; it is stored by the server so can come around through product reviews for example.

Building Exceptional Cyber Security Teams

Maxwell Bond is the recruitment partner of choice in the UK and Germany for tech recruitment across Cyber and Information Security, Cloud & DevOps, Product, Project and Programme, Software Engineering and Development, Salesforce, Design and UX. Contact us today to build and grow high performing technology and digital teams which will drive change, growth, and success within your business.



www.maxwellbond.co.uk

Contact Us

Build high performing teams through exceptional staffing solutions and business advice with Maxwell Bond, the recruitment partner of choice across the UK and Germany.

Jake Adshead
Senior Cyber Security Consultant
jake.adshead@maxwellbond.co.uk
0161 359 3280